

А.С.Цветков

РУКОВОДСТВО ПО РАБОТЕ С КАТАЛОГОМ ТУШО-2

ББК 22.6
Ц27

Рецензенты: докт. физ.-мат. наук, проф. В.В.Витязев
(С.-Петербургский государственный университет)
канд. физ.-мат. наук, научн. сотр. А.Г.Буткевич
(Главная астрономическая обсерватория РАН)

*Печатается по постановлению
Редакционно-издательского совета
С.-Петербургского государственного университета*

Цветков А.С.

Ц27 Руководство по работе с каталогом Tucho-2: учебно-метод.
пособие. – СПб., 2005. – 132 с.

Представлено общее описание звездного каталога Tucho-2. Написаны процедуры чтения каталога и получения статистических характеристик содержащихся в нем данных. Подробно анализируются способы отбора звезд по тем или иным критериям. Рассмотрены методы построения локальных и глобальных карт, демонстрирующих распределение звезд по небесной сфере.

В дополнение к каталогу Tucho-2 рассматривается каталог Tucho-2 Spectral Types. Определены спектральные параллаксы звезд этого каталога. Произведено их сравнение с тригонометрическими параллаксами звезд каталога Hipparcos.

Пособие предназначено для студентов, аспирантов астрономических отделений университетов, а также научных сотрудников, использующих звездные каталоги в своей работе. Все программы написаны на трех языках: Pascal, Fortran и C.

ББК 22.6

© А.С.Цветков, 2006
© С.-Петербургский
государственный
университет, 2006

ОГЛАВЛЕНИЕ

1. Общие сведения о каталоге Tycho-2	4
2. Файлы каталога	8
3. Процедура чтения каталога	13
4. Выборки звезд из каталога Tycho-2	18
5. Общие замечания о построении выборок	21
6. Распределение звезд каталога Tycho-2 на небесной сфере	23
7. Построение локальных карт	31
8. Использование программы RedShift 4	37
9. Каталог Tycho-2 Spectral Type	42
10. Эффективный поиск звезд в каталоге Tycho-2 Spectral Type	51
11. Определение спектральных параллаксов звезд каталога Tycho-2 Spectral Types	56
12. Диаграмма Герцшпрунга-Рессела для звезд каталога Tycho-2 Spectral Types	63
13. Совместное использование данных каталогов Hipparcos и Tycho-2	67
14. Упражнения	74
15. Перечень листингов программ	76
16. Приложение А. Тексты программ на языке C	77
17. Приложение В. Тексты программ на языке Fortran	103
18. Литература	130

1. Общие сведения о каталоге Tycho-2

В 1997 г. Европейское Космическое Агентство (ESA) выпустило в свет звездные каталоги Hipparcos [1] и Tycho, являющиеся результатом космической миссии Hipparcos. Каталог Hipparcos содержит положения, собственные движения и тригонометрические параллаксы более чем 100 000 звезд. Точность положений и параллаксов звезд составляет около 1 мсд¹, собственных движений звезд – 1 мсд/год.

Однако на самом космическом аппарате фактически наблюдалось гораздо больше звезд, но качество их наблюдений (главным образом число измерений за трехлетний период работы спутника на орбите) было гораздо меньшим, чем требовалось для основного каталога. Это обстоятельство позволило непосредственно на материале, полученном на космическом аппарате, построить каталог Tycho, содержащем уже свыше 1 миллиона звезд. Но точность определения положений звезд в этом каталоге была на порядок (от 7 до 25 раз) хуже, чем в каталоге Hipparcos, что резко снизило ценность собственных движений, а практическое использование параллаксов Tycho оказалось просто невозможным.

Сами «сырые» данные, полученные со спутника, содержали сведения еще о полутора миллионах звезд с менее надежными измерениями, которые не вошли в каталоги Hipparcos и Tycho. На основе только этих данных было невозможно получить собственные движения звезд, хотя координаты звезд были определены достаточно надежно. Это обстоятельство побудило коллектив авторов к созданию Tycho-2 – каталога нового поколения. Новизна этого каталога определяется тем, что для вывода собственных движений звезд в качестве первых эпох были использованы обширные ряды наземных астрометрических наблюдений, а для вторых эпох – полный массив данных, собранный космическим аппаратом HIPPARCOS.

В результате появился астрометрический каталог Tycho-2, содержащий положения и собственные движения, а также двухполосную фотометрию 2.5 миллионов звезд (рис. 1), распределенных по всей небесной сфере с плотностью от 25 (у полюсов Галактики) до 150 (у галактического экватора) звезд на квадратный градус (табл. 1). В каталоге приводятся индивидуальные данные для компонент двойных звезд, разделенных более чем на 0.8 угловых секунд. Каталог почти полный до звездной величины 11.5^m, в некоторых областях даже глубже (рис. 2).

¹ мсд – миллисекунда дуги (0.001"), в англоязычной литературе принят термин mas – milli arc second.

Положения и фотометрия звезд Tycho-2 получены из тех же данных, которые использовались при построении каталога Tycho-1² (ESA SP-1200, 1997). Однако каталог Tycho-2 содержит в 2.5 раза больше звезд, по сравнению с Tycho-1, а также является несколько более точным, благодаря улучшенным алгоритмам редукции наблюдений.

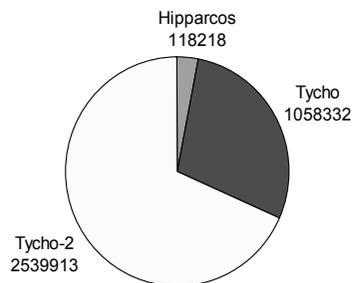


Рис. 1. Число звезд в каталогах Hipparcos, Tycho и Tycho-2.

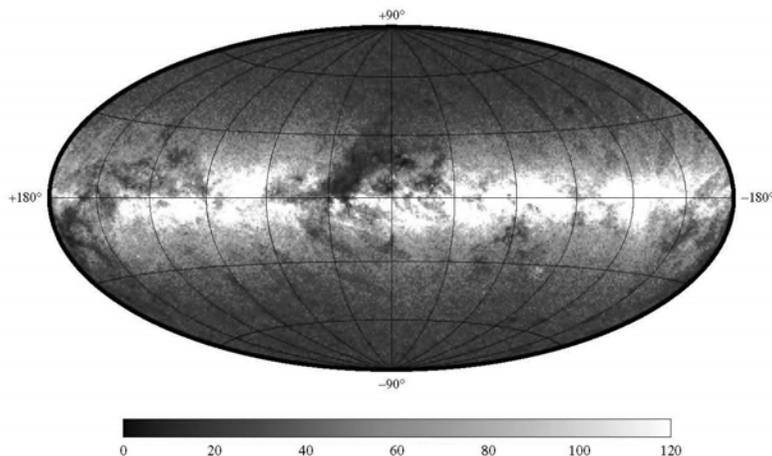


Рис. 2. Плотность распределения звезд каталога Tycho-2 по небесной сфере в галактической системе координат. Единицы измерения – число звезд на квадратный градус.

Особого внимания заслуживают собственные движения звезд Tycho-2. Как уже было сказано, в качестве первой эпохи наблюдений использовались наземные каталоги, прежде всего Astrographic Cata-

² После выхода каталога Tycho-2 каталог Tycho стали обозначать Tycho-1.

logue (AC), а также 143 других наземных каталога, полный список которых можно найти в описании Tycho-2. Всего было использовано 7 700 851 наземное измерение положений звезд. Данные всех этих каталогов были переведены в систему каталога Hipparcos (HICRF³). Достигнутая таким образом точность собственных движений, благодаря значительной по сравнению с каталогом Hipparcos временной базе, составляет около 2.5 мсд/год. Примерно для 100 000 звезд собственные движения определить не удалось, так как для них не нашлось наземных данных, но их положения и фотометрические данные, тем не менее, внесены в каталог.

Каталог Tycho-2 распространяется на компакт-диске [2], содержащем данные каталога, а также документацию в виде основного описания *Guide to the Tycho-2 Catalogue* [3] и двух статей Э. Хёга и др. [4, 5], в которых подробно описывается процедура построения каталога.

Каталог может быть также получен из Интернета через систему VizieR [6], интерактивный атлас Aladin [7] или архив звездных каталогов ESO [8]. Специальная информация, касающаяся Tycho-2, а именно документация, программное обеспечение и возможные исправления, размещена также на официальном сайте Tycho-2 [9]⁴.

В настоящее время (2005 г.) каталог Tycho-2 является самым массовым высокоточным каталогом положений и собственных движений звезд (рис. 3). Он распространяет систему каталога HIPPARCOS (HICRF) на 2.5 миллиона слабых звезд. Это определяет чрезвычайно широкую сферу его применения: от определения относительных координат объектов в фотографических и ПЗС-наблюдениях до исследования звездной кинематики и строения Галактики в ближайших (по видимому, до 2-3 кпк) окрестностях Солнца.

Для практического использования каталогов Hipparcos и Tycho-2 требуются специальные программные средства. К сожалению, в специальной литературе этим вопросам уделяется мало внимания. Для устранения этого недостатка в отношении каталога Hipparcos может служить учебное пособие автора [10]. В настоящем пособии рассматриваются аналогичные задачи применительно к каталогу Tycho-2. После описания обозначений и структуры данных каталога Tycho-2 в пособии приводятся различные процедуры, позволяющие работать с этим каталогом. Основное изложение мы проводим на языке Pascal⁵, а в приложениях даны тексты этих модулей на языках C и FORTRAN-90.

³ Hipparcos International Celestial Reference Frame – реализация ICRS в оптическом диапазоне.

⁴ К сожалению, сайт не обновлялся с 2000 г.

⁵ Использовался свободно распространяемый 32-разрядный компилятор Free Pascal, доступный для DOS, Windows, Linux, FreeBSD и других ОС [11].

Таблица 1. Основные характеристики каталога Tycho-2

Эпоха положений звезд в каталоге Tycho-2	J2000.0
Система каталога	HICRF
Число звезд	2 539 913
Астрометрические ошибки	
положения для звезд с $V_T^6 < 9^m$	7 mas
положения остальных звезд	60 mas
все звезды, собственные движения	2.5 mas/год
Ошибки звездной величины V_T	
для звезд с $V_T < 9^m$	0.013 ^m
остальные звезды	0.10 ^m
Плотность звезд на небесной сфере	
$b = 0^\circ$	150 градус ⁻²
$b = \pm 30^\circ$	50 градус ⁻²
$b = \pm 90^\circ$	25 градус ⁻²
Полнота с вероятностью 90 %	$V_T \sim 11.5$
Полнота с вероятностью 99 %	$V_T \sim 11.0$

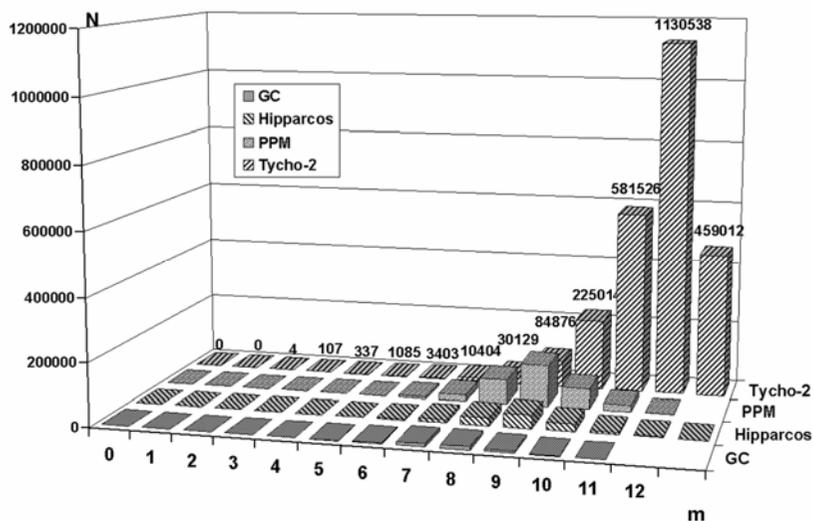


Рис. 3. Распределение звезд каталогов GC, Hipparcos, PPM и Tycho-2 по видимой звездной величине [12].

⁶ V_T – звездная величина в системе Tycho, примерно соответствует m_V (см. с. 10).

2. Файлы каталога

Собственно каталог Tycho-2 состоит из трех файлов и еще одного индексного файла:

- Основной каталог (*catalog.dat*)
- Приложение 1 (*suppl_1.dat*)
- Приложение 2 (*suppl_2.dat*)
- Индекс к Tycho-2 и Приложению 1 (*index.dat*).

Все файлы являются текстовыми ASCII в стандарте DOS, т.е. строка оканчивается двумя символами $\backslashr\n$ (CR, LF) и фактическая длина строки на два байта больше длины текста записи. Все поля разделены символом *вертикальная черта* |, в описаниях формата для ее представления используется фортрановский формат 1X. В связи с этим идентификатор звезды TYC (TYC1, TYC2 и TYC3) составляет одно поле. Пара – номер по каталогу Hipparcos и идентификатор CCDM – также рассматриваются как одно поле. Во всех случаях, придерживаясь соглашений, принятых в каталоге Hipparcos, мы будем обозначать $\sigma_{\alpha^*} \equiv \sigma_{\alpha} \cos \delta$ и $\mu_{\alpha^*} \equiv \mu_{\alpha} \cos \delta$.

2.1. Основной каталог

Основной каталог catalog.dat содержит 2 539 913 записей по 206 байтов (с учетом символов конца строки – 208 байтов). Подробное описание всех полей дано в табл. 2. Однако некоторые из них нуждаются в более подробном комментарии.

Нумерация звезд в Tycho-2 использует систему GSC (Hubble Guide Star Catalog⁷). Собственно идентификатор TYC составляется из трех чисел: номер площадки по GSC – TYC1, номер звезды в площадке – TYC2 и номер компоненты звезды – TYC3 (он обычно равен 1). В этой системе номер звезды принято обозначать тремя числами через дефис, например, 25-14-1.

Для каждой звезды приводятся два вида координат: *средние* положения на J2000.0 и *наблюдаемые* в ходе космического эксперимента положения (приблизительная эпоха – 1991.5). Средние положения представляют собой взвешенные значения координат с учетом собст-

⁷ Работа над этим каталогом велась специальными и добровольными организациями в течение почти 8 лет. Его объем составляет более 1 Гбайта. В GSC приведены координаты и звездные величины почти для 19 миллионов объектов до 14-й, а иногда и 15-й звездной величины. Около 15 миллионов из них являются звездами; почти все остальные – слабые галактики.

Таблица 2. Формат записи файла *catalog.dat*

Байты	Формат	Ед. изм.	Обознач.	Пояснение
1- 4	I4	—	TYC1	[1,9537] TYC1 из TYC или GSC
6- 10	I5	—	TYC2	[1,12121] TYC2 из TYC или GSC
12- 13	I1,1X	—	TYC3	[1,3] TYC3 из TYC
14- 15	A1,1X	—	rflag	[PX] Флаг среднего положения ⁸
16- 28	F12.8,1X	градусы	mRAdeg	[]? α , среднее положение, ICRS, J2000
29- 41	F12.8,1X	градусы	mDEdeg	[]? δ , среднее положение, ICRS, J2000
42- 49	F7.1,1X	мсд/год	pmRA*	[-4418.0,6544.2]?; μ_{α^*} , ICRS, J2000
50- 57	F7.1,1X	мсд/год	pmDE	[-5774.3,10277.3]?; μ_{δ} , ICRS, J2000
58- 61	I3,1X	мсд	e_mRA*	[3,183]?; σ_{α^*} на среднюю эпоху
62- 65	I3,1X	мсд	e_mDE	[1,184]?; σ_{δ} на среднюю эпоху
66- 70	F4.1,1X	мсд/год	e_pmRA*	[0.2,11.5]? $\sigma_{\mu_{\alpha^*}}$
71- 75	F4.1,1X	мсд/год	e_pmDE	[0.2,10.3]? $\sigma_{\mu_{\delta}}$
76- 83	F7.2,1X	год	merRA	[1915.95,1992.53]? ⁹ Средняя эпоха α
84- 91	F7.2,1X	год	merDE	[1911.94,1992.01]? ⁹ Средняя эпоха δ
92- 94	I2,1X	—	Num	[2,36]? Число использованных положений
95- 98	F3.1,1X	—	g_mRA	[0.0,9.9]? Критерий согласия среднего α ¹⁰
99-102	F3.1,1X	—	g_mDE	[0.0,9.9]? Критерий согласия среднего δ
103-106	F3.1,1X	—	g_pmRA	[0.0,9.9]? Критерий согласия μ_{α^*}
107-110	F3.1,1X	—	g_pmDE	[0.0,9.9]? Критерий согласия μ_{δ}
111-117	F6.3,1X	mag	BT	[2.183,16.581]? Звездная величина B_T ¹¹
118-123	F5.3,1X	mag	e_BT	[0.014,1.977]? σ_{B_T}
124-130	F6.3,1X	mag	VT	[1.905,15.193]? Звездная величина V_T
131-136	F5.3,1X	mag	e_VT	[0.009,1.468]? σ_{V_T}
137-140	I3,1X	—	prox	[3,999] Флаг близости ¹²
141-142	A1,1X	—	TYC	[T] Наличие звезды в Tycho-1
143-148	I6	—	HIP	[1,120404]? Номер в каталоге Hipparcos
149-152	A3,1X	—	CCDM	Идентификатор CCDM для звезд Hipparcos
152-165	F12.8,1X	градусы	RAdeg	α , полученная в Tycho-2, ICRS
166-178	F12.8,1X	градусы	DEdeg	δ , полученная в Tycho-2, ICRS
179-183	F4.2,1X	год	epRA	[0.81,2.13] Эпоха–1990.0 для RAdeg
184-188	F4.2,1X	год	epDE	[0.72,2.36] Эпоха–1990.0 для DEdeg
189-194	F5.1,1X	мсд	e_RA*	σ_{α^*} для RAdeg (основанная на модели)
195-200	F5.1,1X	мсд	e_DE	σ_{δ} для DEdeg (основанная на модели)
201-202	A1,1X	—	posflg	[DP] Тип решения в Tycho-2 ¹³
203-206	F4.1	—	corr	Корреляция между наблюдаемыми δ и α^*

⁸ Флаг среднего положения: *пробел* – нормальное среднее положение и собственное движение; P – положение и собственное движение относятся к фотоцентру B_T ; X – среднее положение и собственное движение отсутствует.

⁹ Знак ? означает, что параметр может быть не задан, в этом случае в позициях находятся пробелы.

¹⁰ Критерий согласия – отношение между случайной и модельной ошибкой. Определен только для звезд, у которых Num>2. Значения, превышающие 9.9, усечены до 9.9.

¹¹ Может быть пробел, однако одна из величин B_T либо V_T всегда присутствует.

¹² Расстояние до ближайшего соседнего объекта в Tycho-2 в 0.1".

¹³ *Пробел* – нормальное решение, D – двойная звезда, P – фотоцентр.

венных движений звезд всех использованных каталогов, включая и Tycho-2. Для 4 % звезд нет данных о среднем положении и собственных движений, для таких звезд имеются только наблюдаемые положения. Для близких двойных звезд может случиться такая ситуация, в которой у наземного фотоизображения удастся определить только положение фотоцентра. В этом случае для обеих компонент даются идентичные значения собственных движений, а их качество будет естественно ниже, чем у одиночных звезд.

Переход от средних положений к произвольной эпохе t производится с помощью простой процедуры:

$$\alpha_t = \alpha + (\mu_{\alpha^*} / \cos \delta)(t - 2000.0), \quad (1)$$

$$\delta_t = \delta + \mu_{\delta}(t - 2000.0), \quad (2)$$

где α , δ , μ_{α^*} и μ_{δ} хранятся в байтах 16–57.

Среднеквадратичные ошибки положений вычисляются по значениям, приведенным в байтах 58–75 и средних эпох T_{α} , T_{δ} (байты 76–91):

$$\sigma_{\alpha^*} = \sqrt{\sigma_{\alpha^*}^2 + (T - T_{\alpha})^2 \sigma_{\mu_{\alpha^*}}^2}, \quad (3)$$

$$\sigma_{\delta} = \sqrt{\sigma_{\delta}^2 + (T - T_{\delta})^2 \sigma_{\mu_{\delta}}^2}. \quad (4)$$

Фотометрия в системе Tycho дает голубую и визуальную величины B_T и V_T почти для всех звезд, причем хотя бы одна величина есть у каждой звезды. Не следует слишком доверять этим параметрам для очень слабых звезд ($B_T > 13^m$ или $V_T > 12^m$). Такие значения говорят о том, что звезда настолько слабая, что с трудом поддавалась детектированию в соответствующей полосе пропускания. Шкала звездных величин Tycho не идентична общепринятой шкале Джонсона. Для редукции фотометрических характеристик к шкале Джонсона звезд главной последовательности можно пользоваться приближенными формулами:

$$V = V_T - 0.090(B_T - V_T), \quad (5)$$

$$B - V = 0.850(B_T - V_T). \quad (6)$$

Более подробные сведения можно найти в описании каталогов Hipparcos и Tycho [13].

Считалось, что звезда из Tycho-1 совпадает с объектом в Tycho-2 (отмечено флагом T, байт 141), если различие между положениями в этих каталогах было менее 0.8", кроме такого случая, когда звезда в Tycho-1 имеет флаг астрометрического качества 9 (означающего хорошую фотометрию, но плохую астрометрию). В последнем случае расстояние между объектами, для которых принималось совпадение, увеличивалось до 2.4".

В случае одиночной звезды ее идентификаторы (TYC1, TYC2 и TYC3) будут одинаковыми в обоих каталогах. Если же звезда была атрибутирована как одиночная в Tycho-1, но разрешена в Tycho-2 как близкая пара, то обе ее компоненты помечаются в Tycho-2 флагом T, а идентификатор звезды из Tycho-1 присваивается более яркой компоненте (в полосе V_T); остальным звездам назначаются новые идентификаторы по правилам GSC.

Идентификаторы CCDM¹⁴ приводятся для двойных и кратных звезд каталога Hipparcos, для которых предполагается, что они входят в Tycho-2. При выполнении решения для единого фотоцентра оценивался вклад всех компонент внутри круга диаметром 0.8". Для кратных звезд также полагалось, что их неразрешенные компоненты, заключенные в круге диаметром менее 0.8", есть один объект, и для него определялось положение фотоцентра. Для одиночной звезды предсказанный сигнал от других близких звезд вычитался при анализе подсчета фотонов, таким образом свет от других близких объектов не искажал положение звезды. В случае, когда более чем одна компонента входила в запись Tycho-2, CCDM идентификаторы перечисляют компоненты в алфавитном порядке, независимо от их звездной величины.

В Tycho-2 использовались три различных метода при анализе данных: нормальная процедура для *одиночной звезды*, процедура для *двойной звезды* и процедура для *фотоцентра*. В нормальной процедуре сигнал от близких звезд вычитался, насколько это было возможно. При обработке двойной звезды обе компоненты разделялись и входили в отдельные записи Tycho-2, но в некоторых случаях вторая компонента отбрасывалась. В процедуре для фотоцентра сигнал от близких звезд не вычитался. Эта специальная обработка применялась к известным или подозрительным двойным системам, в которых компоненты надежно не разделялись, в силу чего не мог быть использован алгоритм, применявшийся для двойных звезд.

Процедура определения координат звезд Tycho-2 построена таким образом, что прямое восхождение и склонение могут быть коррелированы. Поэтому в каждой записи приводится коэффициент корреляции между α и δ .

¹⁴ Catalogue of Components of Double and Multiple Stars [14].

2.2. Приложения к каталогу

При составлении Tycho-2 все яркие звезды каталогов Hipparcos и Tycho-1 (ярче $B_T = 2.1$ или $V_T = 1.9$) были исключены. Однако для удобства конечного пользователя эти звезды были вынесены в Приложения. Первое приложение (файл suppl_1.dat) содержит 17 588 звезд с хорошим качеством астрометрических параметров, в то время как второе Приложение (файл suppl_2.dat) содержит 1146 звезд, которые могут быть либо ложными изображениями (так называемые «духи»), либо звездами, изображения которых сильно засвечены близкими яркими звездами. Звезды с флагом астрометрического качества 9 в Tycho-1 исключены из обоих Приложений. В Приложениях присутствуют только звезды, которые есть в каталогах Hipparcos или Tycho-1. Описание полей файлов suppl_1.dat и suppl_2.dat представлено в табл. 3.

Таблица 3. Формат записей файлов suppl_1.dat и suppl_2.dat

Байты	Формат	Ед. изм.	Обознач.	Пояснение
1- 4	I4	—	TYC1	[1,9529] TYC1 из TYC
6- 10	I5	—	TYC2	[1,12112] TYC2 из TYC
12- 13	II,1X	—	TYC3	[1,4] TYC3 из TYC
14- 15	A1,1X	—	flag	[HT] Данные Hipparcos или Tycho-1 ¹⁵
16- 28	F12.8,1X	градусы	mRAdeg	α , ICRS, J1991.25
29- 41	F12.8,1X	градусы	mDEdeg	δ , ICRS, J1991.25
42- 49	F7.1,1X	мсд/год	pmRA*	[]?, μ_{α^*} , ICRS, J1991.25
50- 57	F7.1,1X	мсд/год	pmDE	[]?, μ_{δ} , ICRS, J1991.25
58- 63	F5.1,1X	мсд	e_mRA*	σ_{α^*}
64- 69	F5.1,1X	мсд	e_mDE	σ_{δ}
70- 75	F5.1,1X	мсд/год	e_pmRA*	[]? $\sigma_{\mu_{\alpha^*}}$
76- 81	F5.1,1X	мсд/год	e_pmDE	[]? $\sigma_{\mu_{\delta}}$
82- 83	A1,1X	—	mflag	[BVH] Флаг источника зв. величины ¹⁶
84- 90	F6.3,1X	mag	BT	[2.183,16.581]? Зв. величина Tycho-1 B_T
91- 96	F5.3,1X	mag	e_BT	[]? σ_{B_T}
97-103	F6.3,1X	mag	VT	[]? Tycho-1 V_T или H_p
104-109	F5.3,1X	mag	e_VT	[]? σ_{V_T}
110-113	I3,1X	—	prox	[1,999] Флаг близости
114-115	A1,1X	—	TYC	[T] Наличие звезды в Tycho-1
116-121	I6	—	HIP	[1,120404]? Номер в каталоге Hipparcos
122-122	A1	—	CCDM	Идентификатор CCDM для звезд Hipparcos

2.3. Индексный файл

Этот файл (9538 записей по 42 байта) облегчает поиск звезд в каталоге путем задания соответствия между номерами областей GSC (площадки размером $3.75^\circ \times 3.75^\circ$), по которым отсортированы звезды в каталоге, и номерами звезд.

¹⁵ H – данные каталога Hipparcos и наличие собственных движений; T – данные каталога Tycho-1, в этом случае собственные движения не приводятся.

¹⁶ Пробел – наличие B_T и V_T , B – только B_T , V – только V_T , H – в поле V_T стоит H_p .

3. Процедура чтения каталога

Основой нашего подхода к практическому использованию каталога Tycho-2 является модуль, содержащий процедуру чтения каталога. Полный текст этого модуля приведен в листинге 1. Рассмотрим подробнее его устройство.

В начале модуля идут описания констант, определяющие местоположение каталога и количество звезд в нем. Далее идут битовые константы, которые будут использоваться в условных операторах для выяснения наличия или отсутствия данных о звездной величине. Ниже приводится описание структуры (в терминах языка Pascal – записи) TTycho2. Ее поля полностью соответствуют описанию каталога в табл. 2. Поле magn – целочисленная переменная, установка отдельных битов которой свидетельствует об отсутствии в записи у звезды B_V или V_T .

Модуль экспортирует четыре функции: OpenTycho2 открывает каталог, CloseTycho2 закрывает, ReadTycho2 читает одну запись из каталога, заполняя структуру типа TTycho2, функция DeliveredStars возвращает число прочитанных звезд, либо -1, если каталог не открыт.

Изучим устройство функции ReadTycho2. Функция возвращает значение True в случае успешного чтения и False при достижении конца файла. Такой способ позволяет использовать ее непосредственно в условии оператора цикла While. Оператор Readln считывает одну запись каталога в символьную строку t, в дальнейшем происходит ее синтаксический разбор.

Рассмотрим подробнее эти фрагменты кода на примере чтения идентификатора TYC1:

```
Val(copy(t, 1,4),TYC1,err);
```

Функция copy возвращает символьную строку, состоящую из 4 байтов строки t, начиная с 1-го байта. Эта строка используется в качестве аргумента функции Val, которая преобразовывает текст в своем первом аргументе в числовое значение переменной, задана вторым параметром. Третий параметр – код ошибки конвертации. Он равняется нулю в случае отсутствия ошибок и отличен от нуля, если перевод символьной строки в числовое значение невозможен. Эта особенность используется в интерпретации данных о B_T и V_T , для которых в Tycho-2 не предусмотрено специальных флагов (как, например, pflag), а в случае отсутствия данных в соответствующих полях находятся просто пробелы.

Листинг 1. Модуль чтения каталога Tycho-2

```
Unit Tycho2; { Чтение каталога Tycho2 }

Interface { Интерфейсная часть модуля }

{ Расположение полной версии каталога Tycho2 }
Const Tycho2Name      = 'D:\TYCHO-2\Data\catalog.dat';
      Tycho2NumOfStars = 2539913;
{ Битовые константы флага об отсутствии фотометрии }
Const NoBT = 1; { Нет данных о звездной величине BT }
      NoVT = 2; { Нет данных о звездной величине VT }

Type
TTycho2=record
  { Идентификаторы }
  TYC1 : word; { [1..9537] идентификатор }
  TYC2 : word; { [1..12121] идентификатор }
  TYC3 : word; { [1..3] идентификатор }
  { Средние положения и собственные движения на J2000.0 }
  pflag : char;{ [ PX
                ' ' - нормальное среднее положение и с.д.
                'P' - среднее положение и с.д. фотоцентра,
                'X' - нет среднего положения и с.д. }
  mRAdeg, mDEdeg : double; { Средние RA и DE на J2000.0 }
  pmRA,   pmDE   : double; { Средние с.д. на J2000.0 }
  e_mRA,  e_mDE  : single; { Ошибки RA и DE на сред. эпоху }
  e_pmRA, e_pmDE : single; { Ошибки собственных движений }
  mepRA,  mepDE  : single; { Средняя эпоха RA и DE }
  Num     : word;  { Число использованных положений }
  { Критерии согласия. Действительны для Num>2! }
  g_mRA,  g_mDE  : single; { Критерии согласия средних RA и DE }
  g_pmRA, g_pmDE : single; { Критерии согласия собств. движ. }
  { Фотометрическая информация. Проверять флаг magn }
  BT,     e_BT   : single; { Звездная величина BT и ошибка }
  VT,     e_VT   : single; { Звездная величина VT и ошибка }

  prox           : word;   { [3,999] флаг близости в 0.1" }

  { Дополнительные идентификаторы }
  TYC : char;   { Наличие звезды в Tycho-1 'T' или ' ' }
  HIP : longint; { Номер звезды по Hipparcos или 0 }
  CCDM : string[3]; { Идентиф. CCDM для звезд Hipparcos }

  { Координаты ICRS, полученные на KA }
  RAdeg, DEdeg : double; { RA и DE полученные в Tycho-2 }
  epRA, epDE   : single; { Эпоха-1990.0 для RAdeg, DEdeg }
  e_RA, e_DE   : single; { Ошибки RAdeg, DEdeg, модельные }
  postflg : char; { Тип решения в Tycho2: ' ' - нормальное,
                'D' - двойная звезда, 'P' - фотоцентр }
  corr    : single; { Корреляция между RAdeg и DEdeg }
  magn    : word;   { Битовые флаги отсутствия BT и VT }

end { record } ;
```

```

Procedure OpenTycho2; { Открытие файла каталога }
Procedure CloseTycho2; { Закрытие файла каталога }

{ Чтение одной строки каталога: результат помещается в s,
  возвращает false при достижении конца файла }
Function ReadTycho2(var star : TTycho2):boolean;

Implementation { Исполнительная часть модуля }

var cat : text; { Файловая переменная каталога }

Procedure OpenTycho2;
begin
  FileMode:=0; { Только чтение }
  assign(cat,Tycho2Name); reset(cat);
  position:=0;
end { OpenTycho2 };

Procedure CloseTycho2;
begin
  close(cat);
  position:=-1; { Признак закрытого каталога }
end { CloseTycho2 };

Function ReadTycho2(var star : TTycho2):boolean;
var t : string[206]; { Буфер для чтения строки каталога }
    err : integer; { Код ошибки при преобразовании в число }
begin
{ Если достигнут конец файла, то возвращаем False }
if eof(cat) then ReadTycho2:=false else
begin
  ReadTycho2:=true;
  readln(cat,t);

with star do { Операции с полями структуры star }
begin

  magn:=0; { Обнуление всех битов флага }

  { Процедура Val преобразует текстовую строку в число,
    err отличается от нуля в случае ошибки преобразования.
    Функция copy возвращает подстроку с указанием
    стартовой позиции и количества байтов. }

  { Определение идентификаторов }
  Val(copy(t, 1,4),TYC1,err);
  Val(copy(t, 6,5),TYC2,err);
  Val(copy(t,12,1),TYC3,err);

```

```

{ Определение средних положений и собственных движений }
pflag:=t[14];
if pflag<>'X' then
begin
  { Средние положения }
  Val(copy(t,16,12),mRAdeg,err);
  Val(copy(t,29,12),mDEdeg,err);
  { Собственные движения }
  Val(copy(t,42, 7),pmRA,err);
  Val(copy(t,50, 7),pmDE,err);
  { Ошибки положений }
  Val(copy(t,58, 3),e_mRA,err);
  Val(copy(t,62, 3),e_mDE,err);
  { Ошибки собственных движений }
  Val(copy(t,66, 4),e_pmRA,err);
  Val(copy(t,71, 4),e_pmDE,err);
  { Средняя эпоха RA и DE }
  Val(copy(t,76, 7),merRA,err);
  Val(copy(t,84, 7),merDE,err);
  { Число использованных положений }
  Val(copy(t,92, 2),Num,err);

  { Критерии согласия }
  if Num>2 then
begin
  Val(copy(t, 95,3),g_mRA,err);
  Val(copy(t, 99,3),g_mDE,err);
  Val(copy(t,103,3),g_pmRA,err);
  Val(copy(t,107,3),g_pmDE,err);
end;
end { if pflag };

{ Фотометрия }

{ Звездная величина BT }
Val(copy(t,111,6),BT,err);
if err<>0 then Magn:=Magn or NoBT
  else Val(copy(t,118,5),e_BT,err);

{ Звездная величина VT }
Val(copy(t,124,6),VT,err);
if err<>0 then Magn:=Magn or NoVT
  else Val(copy(t,131,5),e_VT,err);

  { Флаг близости }
  Val(copy(t,137,6),prox,err);

{ Наличие звезды в Tycho-2 }
TYC:=t[141];

{ Номер звезды по Hipparcos или 0 }
Val(copy(t,143,6),HIP,err);

```

```

if err<>0 then begin HIP:=0; CCDM:=''; end
    else CCDM:=copy(t,149,3);

{ RA, DE, полученные в Tycho-2 }
Val(copy(t,153,12),RAdeg,err);
Val(copy(t,166,12),DEdeg,err);

{ Эпохи RA, DE, полученных в Tycho-2 }
Val(copy(t,179,4),epRA,err);
Val(copy(t,184,4),epDE,err);

{ Ошибки RA, DE, полученных в Tycho-2, основанные на модели }
Val(copy(t,189,5),e_RA,err);
Val(copy(t,195,5),e_DE,err);

{ Тип решения в Tycho-2 }
postflg:=t[201];

{ Корреляция между RA и DE }
Val(copy(t,203,4),corr,err);

end { with };

end { if } ;

end { ReadTycho2 };

end.

```

```

Val(copy(t,111,6),BT,err);
if err<>0 then Magn:=Magn or NoBT
    else Val(copy(t,118,5),e_BT,err);

```

В приведенном фрагменте, если возникает ошибка в преобразовании 111-117 байтов строки *t* в число (когда эти байты – пробелы), переменная *err* примет отличное от нуля значение. Обработка этого события заключается в установке флага NoBT в поле Magn (с помощью битовой операции *or*), в противном случае происходит разбор поля *e_BT*.

4. Выборки звезд из каталога Tycho-2

Теперь, с помощью этого модуля можно решать различные задачи. На практике очень часто возникает ситуация, когда требуется произвести некоторый анализ не всех звезд каталога, а только его подмножества, состоящего из звезд, отобранных по одному или нескольким критериям.

Покажем, как это делается на примере задачи о построении распределения звезд каталога Tycho-2 по видимой звездной величине *V*, вычисляемой по формуле (5) с шагом 0.5^m . Эту несложную задачу решает программа на листинге 2. Сформированный этой программой текстовый файл может использоваться в качестве исходных данных при анализе в электронной таблице MS Excel или OO Spread. Построенная таким образом диаграмма изображена на рис. 4.

Рассмотрим подробнее текст программы на листинге 2. С помощью констант Delta, MaxV, MinV задается шаг и диапазон звездных величин. Вычисляемая константа N определяет число элементов массива, в котором будет накапливаться статистика распределения звезд по видимой звездной величине.

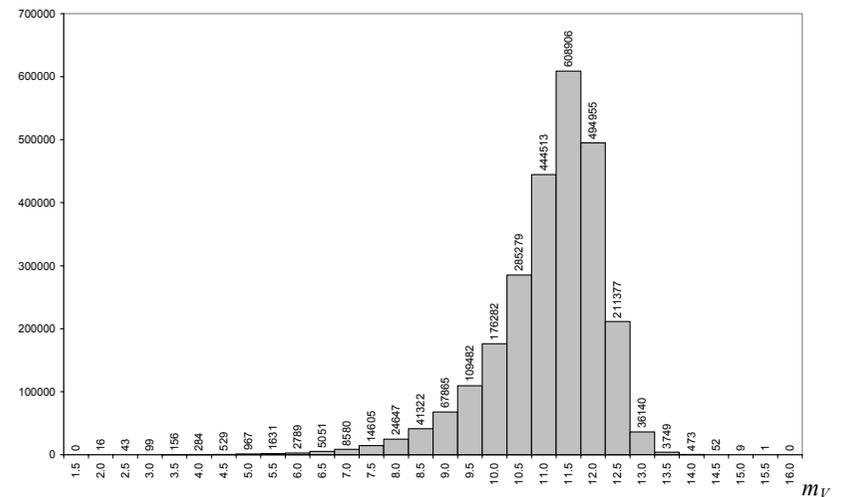


Рис. 4. Распределение звезд Tycho-2 по звездной величине *V*.

Листинг 2. Построение распределения звезд Tycho-2 по видимой звездной величине

```

Program Tycho2_2;
{ Распределение звезд Tycho-2 по звездной величине }

Uses Tycho2;
var star : TTycho2;

const Delta = 0.5; { Шаг по звездной величине }
MinV = -1.0; { Минимальная звездная величина }
MaxV = +16.0; { Максимальная звездная величина }
Res = 'statist.txt'; { Имя выходного файла }
N = round((MaxV-MinV)/0.5); { Число элементов массива }

{ Вспомогательные величины для бегущей полосы прогресса }
k : longint = 0;
step : longint = Tycho2NumOfStars div 80;

var V : single; { Звездная величина }
A : array [0..N] of longint; { Массив статистики }
i : integer; { Индекс элемента }
f : text; { Выходной файл }

begin

for i:=0 to N do A[i]:=0; { Обнуление статистики }

OpenTycho2;

while ReadTycho2(star) do
with star do begin
inc(k); if k mod step = 0 then write('*'); { Полоса прогресса }

{ Пропустить звезду, если нет BT или VT }
if magn<>0 then continue;

V:=(VT-0.090*(BT-VT)); { Вычисление V по формуле (5) }
i:=round((V-MinV)/Delta); { Определение индекса звезды }

{ Увеличение соответствующего элемента массива на единицу }
if (i>=0) and (i<=N) then inc(A[i]);
end;

CloseTycho2;

{ Вывод результатов }
assign(f,Res); rewrite(f);
for i:=0 to N do writeln(f,i*Delta+MinV:6:2,A[i]:7);
close(f);

end.

```

Типизированные константы k и step, которые являются с точки зрения синтаксиса языка Pascal переменными с заданным начальным значением, служат для отображения бегущей полосы из символов '*', позволяющей оценить время вычисления.

Переменная V служит для хранения звездной величины в системе Джонсона. Массив A хранит число звезд, попавших в тот или иной интервал.

Работа программы заключается в переборе всех звезд в цикле while, вычислении видимой звездной величины по формуле (5) и определении номера элемента (индекса) массива, в который попадает звезда. Последнюю задачу решает оператор

$$i := \text{round}((V - \text{MinV}) / \text{Delta}),$$

где функция round производит округление до ближайшего целого. Следуя этой формуле, мы получим, что i-й элемент массива будет увеличен на единицу в том случае, если звездная величина звезды V попадет в интервал

$$[i * \text{Delta} + \text{MinV} - \text{Delta} / 2 \dots i * \text{Delta} + \text{MinV} + \text{Delta} / 2]$$

Оператор

```
if (i>=0) and (i<=N) then inc(A[i]);
```

проверяет, попадает ли i в допустимый интервал индексов массива.

Анализ рис. 4 показывает, что каталог Tycho-2 не содержит самых ярких звезд с $m < 2.25^m$ и звезд слабее 15.75^m . Максимум же распределения приходится на $m = 11.5^m$.

5. Общие замечания о построении выборов

Дадим теперь общие советы, с помощью которых можно легко приспособить основной модуль программы чтения каталога для решения задач, основанных на выборах звезд из каталога.

В рассмотренном примере осуществлялся выбор звезд по видимой звездной величине условными операторами

```
{ Проверка по звездной величине }  
if (s.magn and NoVT)<>0 then continue;  
if (s.VT>=Vmin) and (s.VT<=Vmax) then  
begin  
  . . .
```

Здесь первый оператор `if` проверяет наличие данных о звездной величине V_T . Если данные отсутствуют, то происходит досрочный переход к следующей итерации цикла `while`, в котором происходит обработка каждой звезды. Второй оператор `if` производит собственно проверку нахождения звездной величины в заданном диапазоне. В принципе вид второго оператора можно было изменить так, чтобы его форма была такая же, как и у первого, но в этом случае логическое условие следует поменять на противоположное, а операторные скобки `begin ... end` убрать:

```
if (s.magn and NoVT)<>0 then continue;  
if (s.VT<Vmin) or (s.VT>Vmax) then continue;
```

Стандартное поведение большинства компиляторов предписывает вычисление логических выражений слева направо до тех пор, пока не будет ясен окончательный результат, при этом дальнейшие вычисления выражения прекращаются. Это так называемое неполное вычисление логических выражений позволяет записывать связанные между собой причинным образом условия одним выражением. Таким образом, этот фрагмент может быть записан так:

```
if ((s.magn and NoVT)<>0) or  
    ((s.VT<Vmin) or (s.VT>Vmax)) then continue;
```

В этом операторе часть логического выражения во второй строке не будет выполняться, если первое сравнение окажется истинным, так как этого достаточно для того, чтобы понять, что результат всей операции `or` будет также истинным.

Можно построить условные оператор «классическим» образом, без использования оператора `continue`:

```
if ((s.magn and NoVT)=0) and  
    ((s.VT>=Vmin) and (s.VT<=Vmax)) then  
begin  
  . . .  
end;
```

В такой форме оператора обычно не обойтись без операторных скобок `begin ... end`.

Очевидно, что этот фрагмент программы может быть с легкостью изменен для задания другого критерия отбора или их комбинаций.

Например, зададим критерий отбора звезд с полным собственным движением равным или большим некоторого значения $\mu = \sqrt{\mu_{\alpha^*}^2 + \mu_{\delta}^2}$, при этом будем использовать только те звезды, у которых число наблюдений выше 3. Напомним, что использование собственных движений, средних положений на J2000.0, а также ошибок этих величин возможно только в случае значений поля `pflag` отличного от 'X' (с. 10).

```
if s.pflag<>'X' then  
if (sqrt(sqr(s.pmRA)+sqr(s.pmDE))>mu) and (s.Num>3) then  
begin  
  . . .  
end;
```

Аналогичным образом можно построить самые разнообразные критерии отбора.

6. Распределение звезд каталога Тушо-2 на небесной сфере

В этом и следующем параграфе мы решим некоторые задачи, используя модуль чтения и способ выбора звезд.

Начнем с задачи о построении звездной карты всей небесной сферы для звезд Тушо-2, удовлетворяющих заданному критерию отбора.

Для построения распределения звезд на небесной сфере в последнее время в основном используется проекция Хаммера–Айтофа, она была принята в качестве основной в описании к каталогу Hipparcos. Следующие формулы описывают проекцию Хаммера–Айтофа для сферической галактической системы координат:

$$x = -\frac{2 \cos b \sin \frac{l}{2}}{\sqrt{1 + \cos b \cos \frac{l}{2}}}, \quad l \in [-\pi, +\pi], \quad (7)$$

$$y = -\frac{\sin b}{\sqrt{1 + \cos b \cos \frac{l}{2}}},$$

здесь

l, b – галактические координаты, выраженные в радианах, причем долготу l следует задать в диапазоне $[-\pi, +\pi]$;

x, y – декартовы координаты на плоскости; x будет находиться в диапазоне $[-1, +1]$, а y – в диапазоне $[-2, +2]$, начало координат $(0, 0)$ соответствует направлению на центр Галактики.

Напишем программу, которая строит изображения звезд в этой проекции. В качестве графической платформы используем стандартную библиотеку BGI, реализация которой есть в Borland Pascal, Free Pascal, Borland C++. Она настолько проста, что не составляет труда легко адаптировать программу, использующую BGI, под любую другую систему. Ось X в этом режиме имеет направление слева направо, а ось Y – сверху вниз. Верхний левый угол имеет координаты $(0, 0)$, а правый нижний ($\text{GetMaxX}, \text{GetMaxY}$). Эту систему координат в нашем изложении мы будем называть экранной.

Листинг 3. Процедуры вычисления координат звезд в проекции Хаммера–Айтофа

```

Procedure Aitoff(
    l,b:double; { Сферические координаты в радианах }
    var x,y : double); { Декартовы координаты }
var s : double;
begin
    if l>Pi then l:=l-2*Pi; { Приведение l в диапазон -Pi до +Pi }
    s:=sqrt(1+cos(b)*cos(l/2)); { Знаменатель формул 7 }
    x:=-2*cos(b)*sin(l/2)/s;
    y:=sin(b)/s;
end;

Procedure Screen
(x,y : double; { Декартовы координаты }
X0,Y0: integer;{Экранные координаты начала декартовой системы}
Scale: double;{Масштаб - сколько точек экрана на единицу длины}
var u,v : integer);{ Экранные координаты }
begin
    u:=X0+Round(Scale*x); v:=Y0-Round(Scale*y);
end;

```

Для начала напишем две процедуры Aitoff и Screen. Первая будет переводить сферические галактические координаты в декартовы по формулам (7), а вторая – декартовы в экранные.

Несколько громоздкой может показаться процедура, которая создает рисунок координатной сетки, но никаких алгоритмических сложностей в ней нет.

Если мы хотим построить проекции звезд Тушо-2 в галактической системе координат, то нам необходимо перевести экваториальные координаты, приведенные в каталоге, в галактические. Это можно сделать по формулам

$$b = \arcsin(\sin \delta \cos i - \cos \delta \sin i \sin(\alpha - \varrho)),$$

$$l = \arctg \frac{\sin \delta \sin i + \cos \delta \cos i \sin(\alpha - \varrho)}{\cos \delta \cos(\alpha - \varrho)} + L_0. \quad (8)$$

Здесь

- α, δ – экваториальные координаты;
- $i = 62^\circ 52' 18.295''$ – наклон галактического экватора;
- $\varrho = 282^\circ 51' 34.131''$ – долгота восходящего узла галактического экватора;
- $L_0 = 32^\circ 55' 54.905''$ – долгота направления на центр Галактики;
- l, b – галактические координаты.

Листинг 6 представляет реализацию формул (8).

Листинг 4. Процедура, рисующая координатную сетку

```

function rad(x:double):double; { Перевод градусов в радианы }
begin
  rad:=x/180.0*Pi;
end;

Procedure AitoffGrid
(Step:integer; { Шаг сетки в градусах }
X0,Y0 : integer; { Экранные координаты центра проекции }
Scale : double; { Масштаб - число точек на единицу длины }
Gr : boolean); {Флаг - в градусах или в часах разметка долготы}

var i,j : integer; { Переменные циклов for }
l,b : double; { Галактические координаты }
x,y : double; { Декартовы координаты }
u,v : integer; { Экранные координаты }
s : string[5]; { Строка для подписей }
h : integer; { Для разметки осей }

begin

  { Нанесение сетки меридианов }
  i:=-180; {Первый меридиан слева -180 градусов }

  repeat { Цикл по меридианам }

    l:=rad(i); { Перевод в радианы }
    j:=-90; { Первая точка меридиана }

    repeat { Цикл построения вдоль меридиана }

      { Вычисление точки меридиана }
      b:=rad(j); { Перевод в радианы широты }
      Aitoff(l,b,x,y); { Перевод в декартовы координаты }
      Screen(x,y,X0,Y0,scale,u,v); { Перевод в экранные коорд. }

      { Если точка первая (j=-90), то помещаем графический курсор
        в точку (u,v) функцией MoveTo, если точка не первая, то
        "прочерчиваем" курсором линию из предыдущей точки
        в точку (u,v) функцией LineTo }
      if j=-90 then MoveTo(u,v) else LineTo(u,v);

      j:=j+5; { Шаг 5 градусов обеспечивают гладкий вид меридиана }

    until j>90;

    i:=i+step; { Переход к следующему меридиану }
  until i>180;

```

```

  { Нанесение сетки параллелей - аналогично предыдущему }
  j:=-90;
  repeat { Цикл по параллелям }
    b:=rad(j);
    i:=-180;
    repeat { Цикл построения вдоль параллели }
      l:=rad(i);
      Aitoff(l,b,x,y);
      Screen(x,y,X0,Y0,scale,u,v);
      if i=-180 then MoveTo(u,v) else LineTo(u,v);
      i:=i+5;
    until i>180;
    j:=j+step;
  until j>90;

  { Задание свойств шрифта (может зависеть от граф.библ.) }
  SetTextStyle(DefaultFont,HorizDir,1);
  SetTextJustify(LeftText,BottomText);

  { Подписи меридианов вдоль экватора }
  i:=-180;
  repeat
    { Вычисление координаты точки вывода надписи }
    l:=Rad(i);
    Aitoff(l,0,x,y);
    Screen(x,y,X0,Y0,scale,u,v);
    { Если Gr истина, то разметка в градусах, иначе - в часах }
    if Gr then h:=i
      else begin h:=i div 15; if h<0 then h:=h+24;
      end;
    Str(h,s); { Преобразование значения h в текстовую строку }
    OutTextXY(u+5,v-5,s); { Вывод текстовой строки }

    i:=i+step; { Переход к следующему меридиану }
  until i>180;

  { Изменение выравнивания текста }
  SetTextJustify(RightText,BottomText);

  { Подписи параллелей вдоль нулевого меридиана - аналогично }
  j:=-90+step;
  repeat
    if j<>0 then { Экватор не подписываем }
    begin
      b:=Rad(j);
      Aitoff(0,b,x,y);
      Screen(x,y,X0,Y0,scale,u,v);
      Str(j,s);
      OutTextXY(u-5,v-5,s);
    end;
    j:=j+step;
  until j>90-step;

```

end;

Значения констант даны в системе ISRS, т. е. на равноденствие J2000.0 [15].

Во второй формуле подразумевается *круговой арктангенс* – числитель интерпретируется как синус угла, а знаменатель – как косинус.

В стандартной библиотеке Pascal могут отсутствовать функции arcsin, arctan2, поэтому мы сочли целесообразным также поместить здесь код этих функций.

Листинг 5. Перевод экваториальных координат в галактические

```
Function ArcTan2(x,y:double):double;
var z:double;
begin
  if y<>0 then z:=arctan(x/y)else z:=Pi/2;
  if y<0 then z:=z+Pi;
  ArcTan2:=z;
end;

Function ArcSin(x:double):double;
begin
  ArcSin:=ArcTan2(x,sqrt(1-sqr(x)));
end;

Procedure Galaxy(a,d : double; var l,b : double);
Var sa,ca,sd,cd : double;
Const Leo = 4.936829261; { 282.85948083° }
      L0 = 0.57477039907; { 32.931918056° }
      si = 0.88998807641; { sin 62.871748611° }
      ci = 0.45598379779; { cos 62.871748611° }
begin
  a:=a-Leo;
  sa:=sin(a); ca:=cos(a);
  sd:=sin(d); cd:=cos(d);
  b:=arcsin(sd*ci-cd*si*sa);
  l:=arctan2(sd*si+cd*ci*sa,cd*ca)+L0;
end;
```

Листинг 6. Построение распределения звезд по небесной сфере

```
Program Picture;
Uses CRT,Galactic,Graph,Tycho2,Projection;

Const Vmin=12.0; { Минимальная и максимальная }
      Vmax=13.0; { Звездная величина }
      CX=500; CY=360; { Экранные координаты центра }
      SC=240; { Масштаб перевода декартовых коорд. в экранные }
      Grades = 16; { Число градаций серого в изображении }
var s : TTycho2;
    driver,mode : smallint; { Для инициализации графич. режима }
    color : integer; { Цвет точки }
```

```
RGB : integer; { Для задания цвета палитры }
l, b : double; { Галактические координаты }
x, y : double; { Декартовы координаты }
u, v : integer; { Экранные координаты }
t : string[3]; { Строка для представления числа звезд }
begin
  Driver:=detect; Mode:=0; { Инициализация графического режима }
  InitGraph(Driver,Mode,'D:\BP\BGI');
  SetTextJustify(CenterText,CenterText);

  { Заливка белым цветом всего окна }
  SetRGBPalette(0,255,255,255); { Белый цвет имеет номер 0 }
  SetFillStyle(SolidFill,White); Bar(0,0,GetMaxX,GetMaxY);
  SetRGBPalette(255,0,0,127); { Цвет сетки N255 - темно-синий }
  SetColor(255);
  for color:=1 to Grades do
  begin { Установка палитры из Grades серых цветов }
    RGB:=256-color*(256 div Grades);
    SetRGBPalette(color,RGB,RGB,RGB);
    { Изображение прямоугольника легенды }
    SetFillStyle(SolidFill,Color);
    Bar((color-1)*20,GetMaxY-20,color*20,GetMaxY);
    { Подпись легенды }
    Str(color,t); OutTextXY(color*20-10,GetMaxY-25,t);
  end;
  { Установка остальных цветов палитры в неиспользуемый цвет }
  for color:=Grades+1 to 254 do SetRGBPalette(color,255,0,0);

  OpenTycho2;
  while ReadTycho2(s) do
  begin
    { Проверка по звездной величине }
    if (s.magn and NoVT)<>0 then continue;
    if (s.VT>=Vmin) and (s.VT<=Vmax) then
    begin
      { Перевод в галактические координаты }
      Galaxy(rad(s.RADeg),rad(s.DEDeg),l,b);
      Aitoff(l,b,x,y); { Вычисление декартовых координат }
      Screen(x,y,CX,CY,SC,u,v); { Вычисление экранных координат }
      color:=GetPixel(u,v)+1; { Определение нового цвета пикселя }
      if color>Grades then color:=Grades; { Проверка диапазона }
      PutPixel(u,v,color); { Установка нового цвета пикселя }
    end;
  end; { while }
  CloseTycho2;

  { Рисунок системы координат }
  AitoffGrid(30,CX,CY,SC,True);
  ReadKey; { Ожидание: нажать любую клавишу }
end.
```

Наконец, перейдем к нашей задаче, конкретизировав правила отбора звезд, а именно отобразим на небесной сфере звезды от 12.0 до 13.0 звездной величины.

Текст программы, выполняющей построение распределения звезд по небесной сфере, приведен в листинге 6, а результат ее работы – на рис. 5. Каталог Тучо-2 содержит 2.5 миллиона звезд, что существенно может превышать число точек на экране монитора. Поэтому не достаточно просто поставить для каждой звезды соответствующую точку, возможны ситуации, когда в одну точку будут попадать несколько, возможно десятки звезд. В этом случае следует использовать оттенки (например, серого) цвета для обозначения участков различной плотности. Теперь рисунок создаст более правильное впечатление о распределении звезд по небесной сфере.

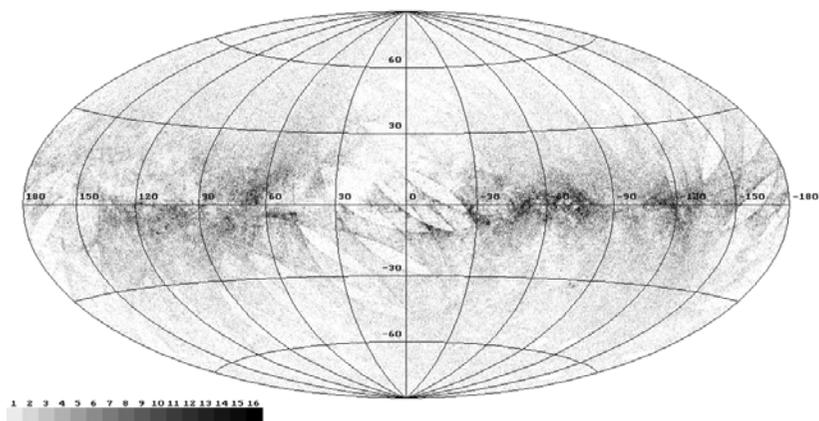


Рис. 5. Плотность распределения звезд Тучо-2 со звездной величиной от 12.0^m до 13.0^m по небесной сфере в галактической системе координат. Шкала указывает число звезд на один пиксел изображения.

В начале программы происходит инициализация графического режима и заливка окна белым цветом.

К сожалению, функции библиотеки BGI¹⁷ работают с цветом только в режиме палитры, т. е. целому числу – номеру цвета – сопоставляются RGB¹⁸-значения цветовых компонент. В нашем случае цвету 0 зада-

ется темно-синий цвет, использующийся для рисования координатной сетки и подписей. Цветам с номерами от 1 до Grades задаются серые оттенки разной плотности от самого светлого (1) до самого темного (Grades). В этом же цикле рисуется легенда плотностей, где цвет соответствует числу звезд, попавших в определенную точку. Остальные цвета палитры устанавливаются в неиспользуемый нами цвет, с тем, чтобы не возникло случайного совпадения заданных серых цветов с другими номерами палитры (это важно в применяемом нами алгоритме).

В основном цикле while происходит чтение данных о звезде, проверка наличия звездной величины V_T и ее значения. Далее происходит преобразование координат в галактические, затем в декартовы координаты проекции и в экранные координаты. Функция GetPixel возвращает существующий цвет пикселя, он увеличивается на единицу, соответственно его цвет становится темнее. Если число звезд, попавших в данную точку, превышает значение Grades, то цвет устанавливается в это значение. Уменьшая значение константы Grades, мы увеличиваем контрастность изображения, но затрудняем исследование областей с высокой звездной плотностью. В каждом случае отбора звезд величину Grades следует подбирать индивидуально.

После завершения основного цикла по верх изображения наносится координатная сетка.

Анализируя рис. 5, на котором представлено решение нашей задачи, легко прийти к выводу о существовании аппаратной селекции звезд, наравне с концентрацией звездной плотности к галактической плоскости.

¹⁷ Borland Graphic Interface – графическая библиотека, использующаяся в Borland Pascal, Borland C++ и других системах.

¹⁸ RGB – цветовая модель, в которой значения красной (Red), зеленой (Green) и синей (Blue) компоненты цвета задаются целыми числами (обычно от 0 до 255).

7. Построение локальных карт

Для идентификации объектов в отдельных участках небесной сферы удобно иметь программное обеспечение, строящее подробные карты звездного неба в небольших площадках. В отличие от каталогов Hipparcos и Tycho-1, для которых создан компакт-диск Celestia 2000, каталог Tycho-2 не имеет штатного программного обеспечения такого рода. Кроме задач идентификации локальные карты могут быть полезными для изучения окрестностей тех или иных объектов.

На рис. 6 представлена такая карта, полученная с помощью программы, текст которой приведен на листинге 8. Для большей информативности звездам различных показателей цвета (или других характеристик) приписан разный цвет, который на черно-белом рисунке отображается разными оттенками серого цвета. Короткие штрихи – отражение собственных движений звезд – показывают в данном случае, насколько звезды сместятся за 5000 лет.

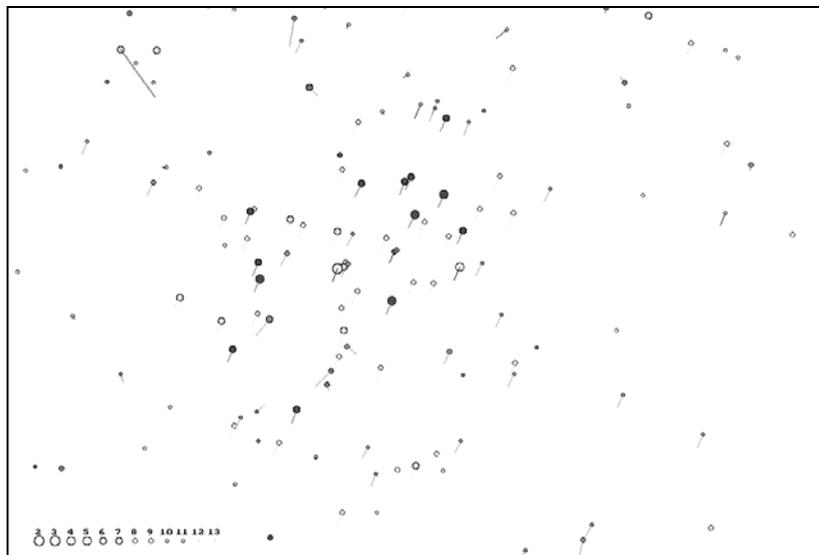


Рис 6. Локальная карта в окрестности звездного скопления M45 «Плеяды». Координаты центра $\alpha=3^h 46^m$, $\delta=23^\circ 58'$, размер по горизонтали – 4° , по вертикали – 3° , север – вверху, штрихи соответствуют смещению звезд за 5000 лет.

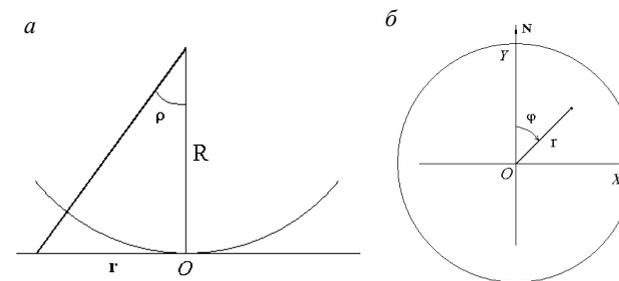


Рис 7. Центральная проекция небесной сферы на плоскость.

Прежде всего, мы должны решить задачу о проекции небесной сферы на плоскость. Так как речь идет о сравнительно малых площадках, можно воспользоваться достаточно простой центральной проекцией. Пусть O – точка касания небесной сферы с тангенциальной (картинной) плоскостью (рис. 7, а), пусть ее экваториальные координаты – A, D (а декартовы координаты – $0, 0$). Если звезда удалена на угловое расстояние ρ от этого направления, то на плоскости она будет удалена на линейное расстояние r от точки касания. Обозначим через φ полярный угол между направлением на северный полюс мира и радиус-вектором звезды r (рис. 7, б). Мы считаем, что наблюдатель смотрит изнутри сферы. В этом случае вычисление полярных координат звезды r и φ для звезды с экваториальными координатами α, δ дается следующими формулами:

$$\cos \rho = \sin D \sin \delta + \cos D \cos \delta \cos(A - \alpha), \quad (9)$$

$$r = R \operatorname{tg} \rho, \quad (10)$$

$$\sin \varphi = \frac{\cos \delta \sin(A - \alpha)}{\sin \rho}, \quad (11)$$

$$\cos \varphi = \frac{\sin \delta - \cos \rho \sin D}{\sin \rho \cos D}. \quad (12)$$

Формулы (9)–(12) реализованы функцией `ToPlane` (листинг 7). В этой функции предполагается, что радиус окружности $R = 1$. В начале функции все углы переводятся в радианную меру и вычисляются все необходимые тригонометрические функции с целью оптимизации дальнейших вычислений.

В условии оператора `if` сравниваются косинусы углов ρ и ρ_{\max} . Если звезда оказывается дальше от точки A, D , чем некоторое максимальное расстояние, дальнейшие вычисления не предпринимаются, и функция сразу возвращает `false`.

Листинг 7. Центральная проекция участка небесной сферы на плоскость

```
Unit Plane;
Interface
Function ToPlane(
  A0,D0 : double; { Координаты центра площадки }
  a, d : double; { Координаты звезды }
  maxr : double; { Максимальное удаление от центра в градусах }
  var x,y: double { Декартовы координаты }
  ):boolean; { Возвращает true в случае успеха }

Implementation

function rad(x:double):double; { Перевод градусов в радианы }
begin
  rad:=x/180.0*Pi;
end;

Function ToPlane(
  A0,D0 : double; { Координаты центра площадки }
  a, d : double; { Координаты звезды }
  maxr : double; { Максимальное удаление от центра в градусах }
  var x,y: double { Декартовы координаты }
  ):boolean; { Возвращает true в случае успеха }

Var SinD0, CosD0, sd,cd,cr,sr,tr,sf,cf,r : double;
begin
  { Перевод всех углов в радианную меру }
  A0:=rad(A0); D0:=rad(D0); a:=rad(a); d:=rad(d);
  { Вычисление синусов и косинусов склонений }
  sd:=sin(d); cd:=cos(d); SinD0:=sin(D0); CosD0:=cos(D0);
  { Вычисление cos(rho) }
  cr:=SinD0*sd+CosD0*cd*cos(A0-A);
  if cr>cos(rad(maxr)) then
  begin { Если звезда ближе к центру, чем maxr }
    { Вычисление sin и tg угла rho }
    sr:=sqrt(1-sqr(cr)); tr:=sr/cr;
    if sr<1.0E-6
    then begin
      sf:=0; cf:=0; { В случае малого rho нет phi }
    end
    else begin { sin и cos позиционного угла phi }
      sf:=(cd*sin(A0-A))/sr;
      cf:=(sd-cr*sinD0)/(sr*cosD0);
    end;
    r:=tr; { Считаем r=1*tr, но возможен и другой множитель }
    X:=r*sf; Y:=r*cf; { Вычисление декартовых координат }
    ToPlane:=True;
  end
  else ToPlane:=False; { Звезда не попадает в поле зрения }
end;

end.
```

Листинг 8. Построение локальной звездной карты

```
Program LocalChart;

Uses CRT,Galactic,Graph,Tycho2,Projection,Plane;

Const A0 = 56.5; D0 = 23.97; { Координаты центра поля зрения }
      R0 = 2.0; { Размер в градусах полуширины экрана }
      Mmin=2; { Минимальная звездная величина }
      Mmax=10.5; { Максимальная звездная величина }
      Time = 5000.0; { лет }

Const CX=500; CY=360;{ Экранные координаты центра }

      k : longint = 0;
      step : longint = Tycho2NumOfStars div 80;

var s : TTycho2;
      driver,mode : smallint; { Для инициализации граф. режима }
      r : integer; { Радиус окружности }
      a1,d1: double; { Координаты на новую эпоху }
      x, y : double; { Декартовы координаты }
      scale: double; { Масштаб }
      u, v : integer; { Экранные координаты }
      t : string[2];{ Для легенды }
      B_V : single; { Показатель цвета }
      color: integer; { Цвет }

begin

  Driver:=detect; Mode:=0; { Инициализация графического режима }
  InitGraph(Driver,Mode,'D:\BP\BGI');
  SetTextJustify(JustifyCenter,JustifyCenter);
  { Заливка белым цветом всего окна }
  SetFillStyle(SolidFill,White); Bar(0,0,GetMaxX,GetMaxY);

  SetColor(Black);
  for r:=2 to 13 do { Легенда карты - звездные величины }
  begin
    circle(r*20,GetMaxY-20,7 - r div 2);
    Str(r,t);
    OutTextXY(r*20,GetMaxY-30,t);
  end;

  { Масштаб - половина ширины окна есть угловое расстояние }
  scale:=(GetMaxX div 2)/(sin(rad(R0))/cos(rad(R0)));

  OpenTycho2;
```

```

while ReadTycho2(s) do
begin
inc(k); if k mod step = 0 then write('*'); { Бегущая полоса }

{ Проверка по звездной величине }
if ((s.magn and NoVT)<>0) or ((s.magn and NoBT)<>0)
then continue;
{ Допустимый диапазон }
if (s.VT<Mmin) or (s.VT>Mmax) then continue;

{ Проектирование на плоскость }
if ToPlane(A0,D0,s.RAdeg,s.DEdeg,R0,x,y) then
begin
Screen(x,y,CX,CY,scale,u,v); { Экранные координаты }
r:=7-Round(s.VT/2); if r<1 then r:=1;
B_V:=0.850*(s.BT-s.VT); { Вычисление B-V по формуле 6 }

if (B_V<-0.5) { Выбор цвета }
then color:=LightBlue
else if (B_V<0.0)
then color:=LightCyan
else if (B_V<0.5)
then color:=Yellow
else if (B_V<1.0)
then color:=LightRed
else color:=Red;

{ Изображение звезды - цветной кружок с черной границей }
SetColor(Black); SetFillStyle(SolidFill,color);
FillEllipse(u,v,r,r);

if s.pflag<>'X' then { Есть данные о собственном движении }
begin
{ Вычисление положения звезды через Time лет }
a1:=s.RAdeg+(s.pmRA*0.001/3600.0)/cos(rad(s.DEdeg))*Time;
d1:=s.DEdeg+(s.pmDE*0.001/3600.0)*Time;
if ToPlane(A0,D0,a1,d1,R0,x,y) then
begin
{ Проводим отрезок из старого положения звезды в новое }
SetColor(Color); MoveTo(u,v);
Screen(x,y,CX,CY,scale,u,v);
LineTo(u,v);
end;
end; { if pflag }

end; { if ToPlane 1 }
end; { While }

CloseTycho2;
Writeln('Ok');
ReadKey;
end.

```

В случае очень малых ρ (т. е. звезда находится очень близко к центру поля зрения) возможны большие ошибки при вычислении по формулам (11), (12), поэтому в программе предусмотрена проверка этого факта с целью избежать деления на очень малые значения, включая и нуль.

Так как угол φ в формулах (11), (12) отсчитывается по часовой стрелке от направления на северный полюс мира, то декартовы координаты проекций звезд на касательной плоскости вычисляются по следующим формулам:

$$x = r \sin \varphi, \quad y = r \cos \varphi. \quad (13)$$

Следующие отсюда развернутые формулы для тангенциальных координат x, y приведены в монографии А.А. Киселева [16].

Рассмотрим теперь текст программы LocalChart. Константы A_0, D_0 задают направление в градусах на центр поля зрения; константа R_0 – охват в градусах половины горизонтального размера экрана. В данном случае по ширине изображение имеет размер 4° . Константы R_{\min} и R_{\max} определяют диапазон отображаемых звездных величин V_T . Для отображения собственных движений в виде штрихов вычисляются новые положения звезд на эпоху $J2000.0+Time$.

После инициализации графического окна цикл `for` отображает легенду карты, рисуя окружности разных диаметров для звезд разных величин.

Далее мы определяем масштабный множитель `scale` по формуле

$$\frac{X_{\max}/2}{\operatorname{tg} R_0}, \quad (14)$$

здесь X_{\max} – разрешение экрана в точках по горизонтали, R_0 – значение константы R_0 .

В начале тела цикла `while` производится отбор звезд по величине V_T , затем звезда проецируется на плоскость. В случае успеха (звезда попадает в площадку: $\rho < R_0$) процедура `Screen` вычисляет экранные координаты u, v , затем определяется радиус кружка, соответствующий звездной величине, а также его цвет в зависимости от показателя цвета.

Процедура `FillEllipse` рисует закрашенный цветом `color` кружок с черной границей.

По формулам (5) и (6) мы можем получить координаты звезды a_1, d_1 на эпоху $J2000.0+Time$. Эти координаты также проецируются на плоскость, и мы проводим отрезок из центра кружка к точке, соответствующей новым координатам звезды.

8. Использование программы RedShift 4

Компьютерный планетарий RedShift (компания Maris Technologies) – один из самых известных в астрономическом мире программных продуктов. Первая версия появилась в 1993 г. С тех пор RedShift считается признанным лидером в области моделирования астрономических явлений. Программа создается коллективом в основном отечественных авторов в городе Королев [16]. Несмотря на это, существует только одна русская версия программы (3-я версия), которая вышла лишь в 1998 г.

В четвертой версии программы, появившейся в 2000 г., в качестве базового каталога применяется Tucho-2, а для звезд слабее 11.5^m программа использует комбинацию каталогов 4M и Hubble Guide Star Catalog (HGSC). Осуществлена идентификация звезд базового каталога с каталогами переменных звезд General Catalog of Variable Stars (GCVS) и Catalog of New Suspected Variables (NSV), с каталогами двойных и кратных звезд Hipparcos Double and Multiple Systems Annex (HDMSA) и Washington Double Star Catalog (WDS).

Кроме звезд программа содержит:

- теорию движения планет и 63 спутников;
- базу данных комет (1 700) и астероидов (15 000);
- 73 000 галактик из каталога Catalogue of Principal Galaxies;
- 5 000 туманностей всех типов;
- более тысячи звездных скоплений.

В силу этого программа, несмотря на свое первоначальное предназначение как мультимедиа-энциклопедии, вполне может использоваться для решения некоторого круга прикладных астрономических задач: построение идентификационных карт для обработки фотографических и ПЗС-наблюдений, получения весьма подробной информации об отдельных звездах Tucho-2, включая астрометрические и астрофизические характеристики, вычисление восходов и заходов, горизонтальных координат, видимых мест звезд на произвольную эпоху и место наблюдения, проводить кросс-идентификацию отдельных звезд в различных каталогах. К сожалению, программа не обладает способностями делать выборки звезд из Tucho-2 по тем или иным критериям.

Рассмотрим основные этапы работы с программой RedShift 4.

При первом запуске программы следует установить координаты места наблюдения, вызвав диалог из главного меню программы Navigation – Set Home Location (рис. 8). В панели Navigation можно установить время наблюдения и направление на объект, а также желаемый

масштаб изображения (рис. 9). После этих установок главное окно программы будет выглядеть, как на рис. 10.

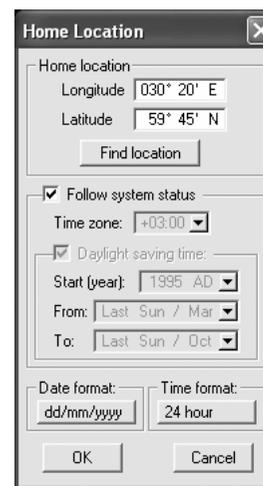


Рис. 8. Установка Home Location.

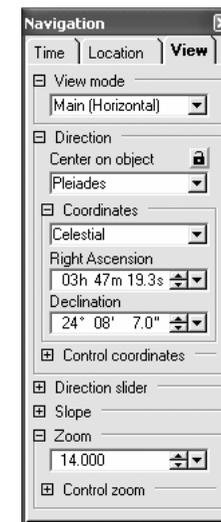


Рис. 9. Панель Navigation.

Выделение пункта Stars в панели Tool Panel приведет к раскрытию диалогового окна из четырех вкладок, позволяющего управлять отображением звезд в основном окне программы. На вкладке General можно управлять предельными звездными величинами, способами пометки двойных и кратных звезд и подписями (рис. 11). Вкладка Binaries определяет предельное расстояние между компонентами, при котором двойная звезда отображается специальным символом. Если расстояние между компонентами звезды больше этого предельного расстояния, то программа отображает непосредственно точками каждую компоненту. Вкладка Variables позволяет определить вид маркера для каждого типа переменности. Вкладка Labels позволяет задать вид меток у звезд. Например, можно задать режим отображения номеров звезд по каталогу Tucho-2 (рис. 12).

Щелчок основной клавишей мыши по объекту позволяет просмотреть его свойства: видимость объекта, время восхода-захода, имена и номера по разным каталогам (рис. 13), свойства звезды (рис. 14), тип переменности (рис. 15).

На этом интересные для нас возможности программы RedShift 4 исчерпываются.

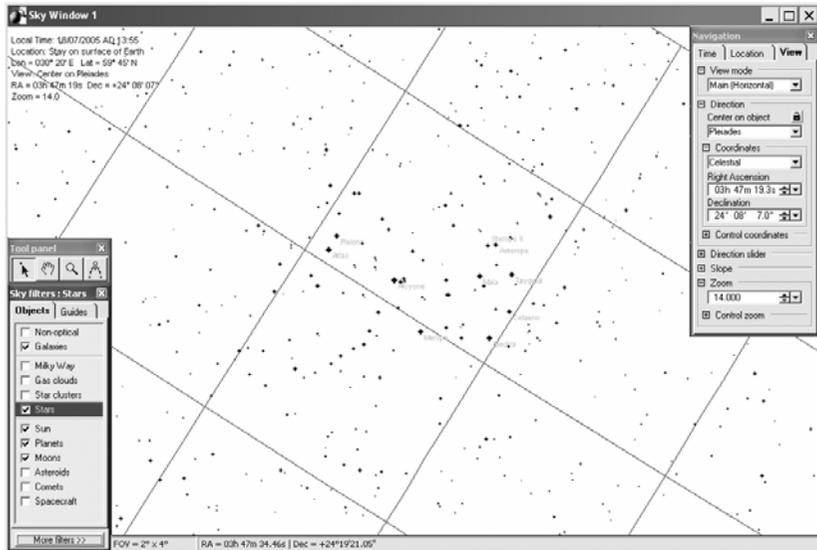


Рис. 10. Общий вид главного окна программы RedShift 4.

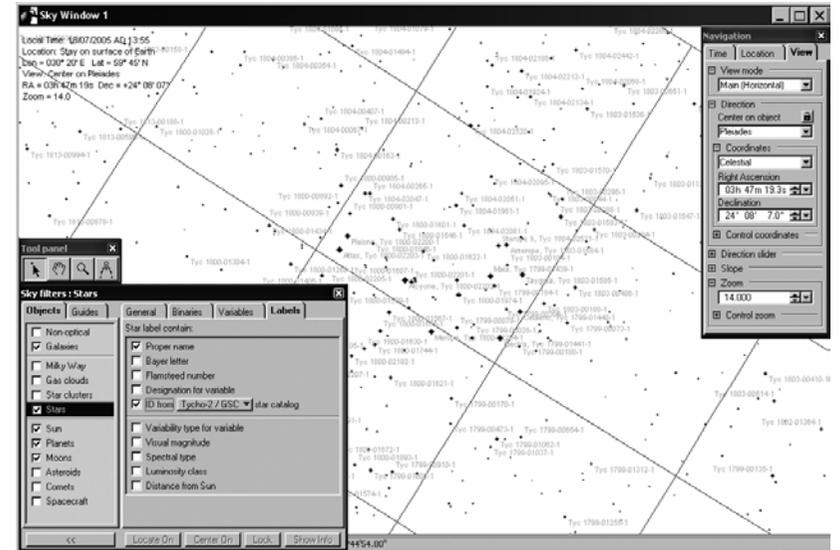


Рис. 12. Включено отображение идентификаторов звезд по Tycho-2.

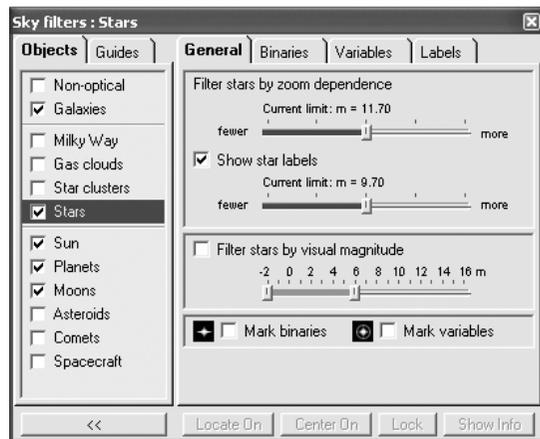


Рис. 11. Вкладка General в установке свойств отображения звезд.

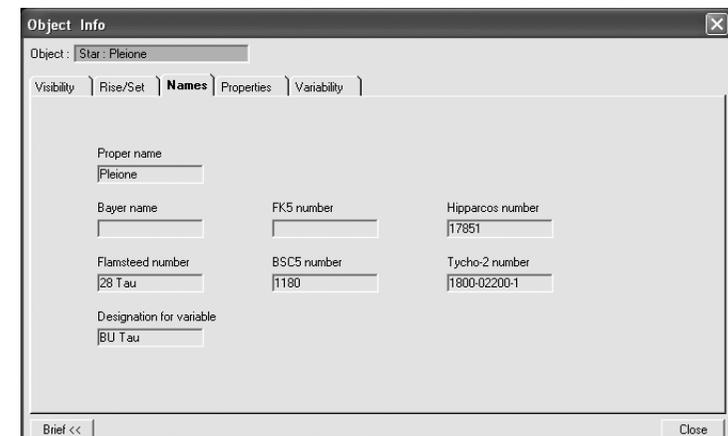


Рис. 13. Номера звезды по разным каталогам.

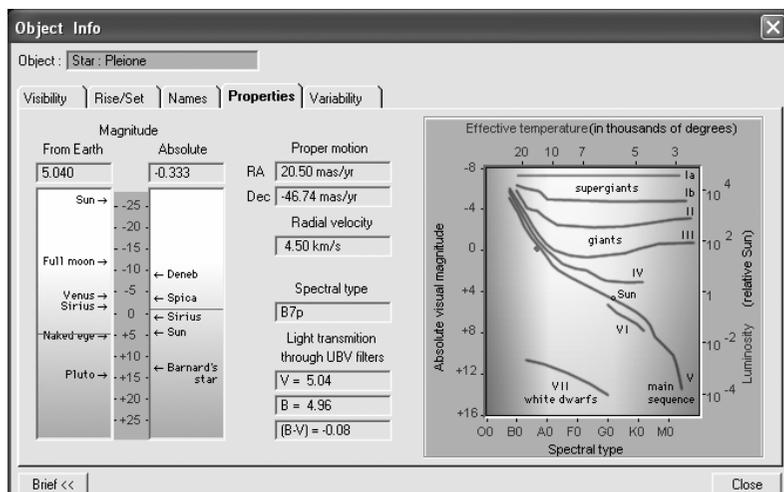


Рис. 14. Собственные движения, лучевая скорость, спектральный класс и фотометрия.

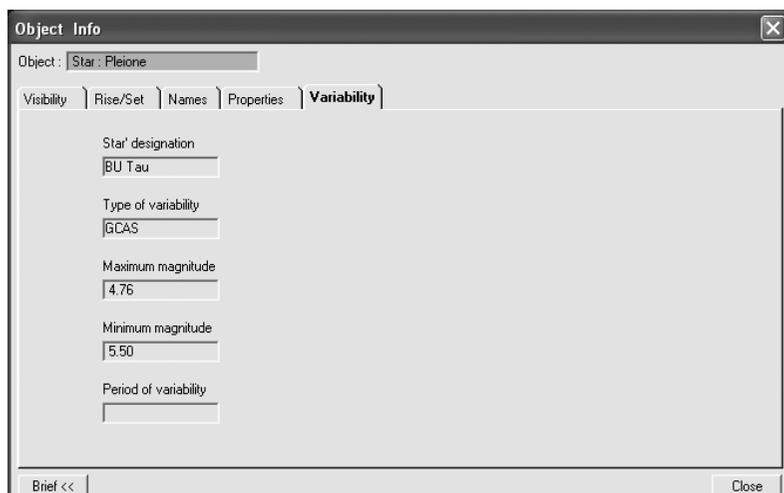


Рис. 15. Тип переменности, амплитуда и период.

9. Каталог Tycho-2 Spectral Type

К большому сожалению, каталог Tycho-2 не содержит ни параллаксов, ни каких-либо данных о светимостях звезд, по которым можно было бы судить о расстояниях до них. Отчасти этот недостаток может восполнить Tycho-2 Spectral Type Catalog [18]. Файлы этого каталога доступны через систему CDS [19]. Помимо развернутых данных о спектральных классах в каталоге приведены классы светимости, температурные подклассы, эффективные температуры.

Авторами каталога была проведена кросс-идентификация 351 863 звезд каталога Tycho-2 с каталогами, содержащими данные о спектральных классах звезд. В их число вошли Мичиганский двумерный каталог спектральных классов для звезд HD [20], каталог звездных спектров в системе Моргана–Кинана [21], каталог «Расширение классификации МК» [22]; а также ряд астрометрических каталогов, которые содержали информацию о спектральном классе звезды: FK5 [23], North PPM и South PPM [24].

Интересно, что для взаимной идентификации звезд использовалась интернет-система VizieR в сочетании с астрономической базой данных SIMBAD, размещенной на сервере CDS в Страсбурге (Франция). В качестве критерия совпадения звезд было использовано значение 10" для каталогов спектральных классов и 1" для астрометрических каталогов. Следует сказать, что 342 986 (97.5 %) звезд имеют согласие по координатам в пределах 1", и только 471 (0.13 %) звезда имеет расхождение более 5".

Кроме совпадения координат звезд изучалась разность их визуальных звездных величин из каталога Tycho-2 и каталога спектрального класса. В каждом случае использовалась процедура приведения звездной величины к шкале Джонсона. В итоге среднеквадратичное расхождение блеска для всех звезд составило 0.40^m. Большинство всех звезд (97.5 %) имели разность в пределах 1^m, а 99.8 % в пределах 2^m. Случаи резкого расхождения положения или блеска приведены в приложении к каталогу.

Так как каталог Tycho-2 Spectral Types предназначен прежде всего для массовых исследований, а обозначения спектральных классов бывают порой весьма длинными и громоздкими, то была предложена схема упрощенного представления спектрального класса с помощью трех однобайтовых полей TClass – температурный класс (O, B, A и т. д.), SClass – подкласс, и LClass – класс светимости. Они представляют собой переформатированное и упрощенное представление спектрального класса, удобное для компьютерной обработки. Например, если исходный спектральный класс был K2/3 III, он станет K2 3, где

К – класс температуры, 2 – подкласс, 3 – класс светимости. Если данные о классе светимости или температурном подклассе не известны, то вместо значений в записи звезды будут пробелы.

Если обозначение класса более сложное, то, конечно, происходит потеря тонкой спектральной классификации, которая не очень важна при массовых обработках. Примеры: исходные классы A9/F2 V, B2.5 V и G8 IV/V станут обозначаться A9 5, B2 5 и G8 4.

Однако авторы каталога сохранили возможность работы с индивидуальными звездами, включив в каталог спектральную классификацию в «классическом» обозначении, в таком виде, как в каталоге-источнике спектрального класса.

Каталог Tucho-2 Spectral Types состоит из трех файлов:

- Основной каталог (файл *catalog.dat*)
- Приложение А (файл *appdx.a.dat*)
- Приложение В (файл *appdx.b.dat*)

Основной каталог содержит 351 863 записи по 124 байта (с учетом символов конца строки – 126 байтов). Описание всех полей дано в табл. 4. В отличие от каталога Tucho-2 в этом каталоге в качестве разделителей полей используются пробелы.

Поля TYC1, TYC2, TYC3 – стандартные идентификаторы Tucho-2. Поля mRAdeg, mDEdeg – средние координаты звезды на эпоху J2000.0. VT и BT – звездные величины V_T и B_T , полученные на аппарате Hipparcos. Значение 99.99 указывает на неопределенность этих величин.

Трехбайтовое поле r_SpType указывает на источник спектрального класса:

mc1 – Michigan Catalog, Vol. 1, III/31; k83 – Kennedy 1983, III/78;
 mc2 – Michigan Catalog, Vol. 2, III/51; fl – FK5, Part I, I/149;
 mc3 – Michigan Catalog, Vol. 3, III/80; fII – FK5, Part II, I/175;
 mc4 – Michigan Catalog, Vol. 4, III/133; ppN – PPM North, I/146;
 mc5 – Michigan Catalog, Vol. 5, III/214; ppS – PPM South, I/193;
 j64 – Jaschek et al. 1964, III/18; sim – SIMBAD Astronomical Database.

Альтернативное обозначение звезды Name – обозначение, взятое из источника спектрального класса. Важный параметр Dist – расстояние между объектом в каталоге Tucho-2 и соответствующим объектом в каталоге спектральных классов.

Таблица 4. Формат записей файла *catalog.dat* каталога Tucho-2 Spectral Type

Байты	Формат	Ед. изм.	Обознач.	Пояснение
1- 3	A3	—	—	Метка, содержащая текст TYC
5- 8	I4	—	TYC1	Первый идентификатор Tucho-2
10- 14	I5	—	TYC2	Второй идентификатор Tucho-2
16	I1	—	TYC3	Третий идентификатор Tucho-2
18- 29	F12.8	градусы	mRAdeg	α , среднее положение, J2000
31- 42	F12.8	градусы	mDEdeg	δ , среднее положение, J2000
44- 49	F6.3	mag	VTmag	Звездная величина V_T (99.99 – не определ.)
51- 56	F6.3	mag	BTmag	Звездная величина B_T (99.99 – не определ.)
58- 60	A3	—	r_SpType	Источник спектрального класса
62- 76	A15	—	Name	Дополнительное обозначение звезды
78- 83	F6.3	сек. дуги	Dist	Расстояние между объектом Tucho-2 и объектом из каталога спектрального класса
85- 90	F6.2	mag	Mag	Звездная величина из каталога спектрального класса (99.99 – не определено)
92	A1	—	f_Mag	[VPBX*] Тип звездной величины
94	A1	—	TClass	Температурный класс
95	I1	—	SClass	Температурный подкласс (возможен пробел)
97	I1	—	LClass	Класс светимости (возможен пробел)
99-103	I5	K	Teff	Эффективная температура, основанная на спектральном классе
105-124	A20	—	SpType	Спектральный класс

Поле Mag определяет звездную величину, взятую в каталоге спектральных классов (значение 99.99 также обозначает неопределенность этого поля), а f_Mag – ее тип:

V – визуальная, X – неизвестная,
 P – фотографическая, * – нет данных о звездной величине.
 B – в полосе B,

О полях TClass, SClass и LClass уже говорилось ранее. Поле SpType хранит текстовое обозначение спектральной классификации, взятое из источника.

Поле Teff вычислено на основе таблиц [25], позволяющих перевести спектральный класс в эффективную температуру. В случае одномерной спектральной классификации (не указан класс светимости) предполагался класс светимости V для звезд ранее G5, III для G5-K9 и I для M-звезд. Если отсутствовал и подкласс, то он предполагался равным 5. И, наконец, считалось, что R-звезды имеют свойства K-звезд; предполагалось, что N-, S-, C- и P-звезды, имеют температурные свойства M-звезд, а звезды Вольфа-Райе W – температуры O-звезд.

При кросс-идентификации звезд в Tucho-2 и звезд в астрофизических каталогах в 104 случаях возникли серьезные трудности. Эти объекты перечислены в Приложении А (формат файла представлен

в табл. 5). Значение поля Flag может быть либо d, что означает расхождение более 5" в положении; либо m, что говорит о значительной разнице (свыше 2^m) в звездной величине объекта в Tycho-2 и астрофизическом каталоге.

Приложение В включает 111 408 дополнительных звезд Tycho-2, которые не вошли в основной каталог. Эти объекты были взяты из каталога Фабрициуса [26] на основе совпадения идентификатора HD. В отличие от основного каталога Приложение содержит лишь одномерную спектральную классификацию. Формат записей каталога представлен в табл. 6.

Таблица 5. Описание формата файла *appdxa.dat* каталога *Tycho-2 Spectral Type*

Байты	Формат	Ед. изм.	Обознач.	Пояснение
1- 3	A3	—	—	Метка, содержащая текст TYC
5- 8	14	—	TYC1	Первый идентификатор Tycho-2
10- 14	15	—	TYC2	Второй идентификатор Tycho-2
16	11	—	TYC3	Третий идентификатор Tycho-2
19	A1	—	Flag	[dm] Флаг неопределенности

Таблица 6. Формат записей файла *appdxb.dat* каталога *Tycho-2 Spectral Type*

Байты	Формат	Ед. изм.	Обознач.	Пояснение
1- 4	14	—	TYC1	Первый идентификатор Tycho-2
6-10	15	—	TYC2	Второй идентификатор Tycho-2
12	11	—	TYC3	Третий идентификатор Tycho-2
13	A1	—	f TYC1	[S] Установлен для звезд Tycho-2 Supp. 1
15-20	16	—	HD	Идентификатор Henry Draper
22-24	A3	—	SpType	Одномерный спектр. класс из каталога HD
26	11	—	n HD	Число HD звезд для этой звезды Tycho-2
28	11	—	n TYC1	Число звезд Tycho-2 для звезды HD
30	A1	—	TDSC	[D] звезда в Tycho-2 Double Star Catalog
32	A1	—	Rem	[DEGMR*] Комментарии из HD
34	A1	—	[N]	Нет данных в notes.dat каталога Фабрициуса
36-40	15	K	Temp	Оценка температуры по спектральному классу (значение -9999 – не определено)

Поле Rem может принимать следующие значения:

- D – запись удалена в печатной версии каталога
- E – изображение на краю пластинки
- G – положение и BD номер взяты непосредственно из AGK1
- M – использовано множественное изображение
- R – есть комментарии в печатной версии каталога
- * – спектральный класс относится к скоплению, туманности и т. п.

Текст модуля, читающий данные Tycho-2 Spectral Type, приведен в листинге 9. Хотя по значению полей *VTmag*, *VTmag*, *Mag*, *SClass* и

LClass можно судить об их неопределенности, мы традиционно включили массив битов и битовые константы, позволяющие узнать об определенности тех или иных параметров, не прибегая к анализу их значений.

В качестве примера использования процедуры чтения напишем программу, позволяющую узнать, сколько звезд в каталоге имеют данные о классе светимости и как звезды с классами светимости распределены по спектральным классам. Эта важная задача, поскольку мы можем оценить, для какого числа звезд можно приблизительно оценить расстояния методом спектральных параллакс. Текст этой программы представлен на листинге 10, а обработанные в электронной таблице результаты – на рис. 16 и 17.

Для накопления статистических данных в программе используются два массива: одномерный массив *Lum*, который будет содержать распределение звезд по классам светимости, и двумерный массив *TwoDim* для подразделения звезд на спектральные классы внутри классов светимости. Переменная *NoLum* предназначена для хранения числа звезд, не имеющих данных о спектральном классе.

В теле цикла *while* с помощью операторе *case* определяется индекс *i* на основе данных о спектральном классе, при этом мы считаем общую статистику для звезд класса *B* и редких классов *W*, *O*. Так же поступаем с классами *K*, *R* и *M*, *N*, *S*, *C*, *P*. Индекс *j* вычисляется путем вычитания из кода символа *LClass* кода символа '0'.

Результат выводится в текстовый файл, сначала одномерная таблица, затем двумерная. Графическое представление наших вычислений показано на рис. 16 и 17.

Из рис. 16 мы заключаем, что только 165 039 звезд имеют данные о классе светимости, в то время как 186 824 звезд этих данных не имеют.

Анализ рис. 17 показывает, что среди звезд класса светимости III наиболее распространенными являются красные гиганты спектрального класса *K* (34 428 звезд) и звезды класса *G* (12 231 звезда). Среди звезд главной последовательности наиболее мощными популяциями являются звезды *F*, *A* и *G* (29 622, 18 217, 13 942 звезды соответственно).

Листинг 9. Чтение каталога Tycho-2 Spectral Types

```

Unit Tycho2Sp; { Чтение каталога Tycho2 Spectral Types }
Interface { Интерфейсная часть модуля }
{ Расположение полной версии каталога Tycho2 Spectral Types }
Const Tycho2SPName = 'D:\TYCHO2SPT\catalog.dat';
Tycho2SPNumOfStars = 351863;
{ Битовые константы флага о наличии информации }
NoBT = 1; { Нет данных о звездной величине BT }
NoVT = 2; { Нет данных о звездной величине VT }
NoMag = 4; { Нет данных о звездной величине Mag }
NoS = 8; { Нет данных о спектральном подклассе }
NoL = 16; { Нет данных о спектральном подклассе }

Type TTycho2SP=record
TYC1 : word; { Идентификатор TYC1 }
TYC2 : word; { Идентификатор TYC2 }
TYC3 : word; { Идентификатор TYC3 }
mRAdeg, mDEdeg : double; { Средние RA и Dec на J2000.0 }
VTmag, BTmag : single; { Звездные величины VT и BT }
r_SpType : string[3]; { Источник спектрального класса }
Name : string[15]; { Дополнительное обозначение звезды }
Dist : single; { Расстояние между объектом Tycho-2 и
объектом из каталога спектрального класса }
Mag : single; { Звездная величина из каталога
спектрального класса }
f_Mag : char; { Тип звездной величины [VPBX*] }
TClass : char; { Температурный класс }
SClass : char; { Температурный подкласс }
LClass : char; { Класс светимости }
Teff : longint; { Эффективная температура }
SpType : string[20]; { Спектральный класс из источника }
info : word; { Битовые флаги отсутствия информации }
end;

Procedure OpenTycho2Sp; { Открытие файла каталога }
Procedure CloseTycho2Sp; { Закрытие файла каталога }
{ Чтение одной строки каталога: Результат помещается в star,
возвращает false при достижении конца файла }
Function ReadTycho2Sp(var star : TTycho2Sp):boolean;

Implementation { Исполнительная часть модуля }
var f : text; { Файловая переменная каталога }

Procedure OpenTycho2Sp;
begin
FileMode:=0; { Только чтение }
assign(f,Tycho2SPName); reset(f);
end { OpenTycho2Sp };

Procedure CloseTycho2Sp;
begin
close(f);
end { CloseTycho2Sp };

```

```

Function ReadTycho2Sp(var star : TTycho2Sp):boolean;
var t : string[124]; { Буфер для чтения строки каталога }
err : integer; { Код ошибки при преобразовании в число }
begin
{ Если достигнут конец файла, то возвращаем False }
if eof(f) then ReadTycho2Sp:=false else
begin
readln(f,t); ReadTycho2Sp:=True;
with star do { Операции с полями структуры star }
begin
info:=0; { Обнуление всех битов флага }
{ Определение идентификаторов }
Val(copy(t, 5,4),TYC1,err);
Val(copy(t,10,5),TYC2,err);
Val(copy(t,16,1),TYC3,err);

{ Определение средних положений }
Val(copy(t,18,12),mRAdeg,err);
Val(copy(t,31,12),mDEdeg,err);

{ Звездные величины VTmag и BTmag }
Val(copy(t,44,6),VTmag,err);
if VTmag>99.9 then Info:=Info or NoVT;
Val(copy(t,51,6),BTmag,err);
if BTmag>99.9 then Info:=Info or NoBT;

{ Источник спектрального класса }
r_SpType:=copy(t,58,3);
{ Дополнительное имя звезды }
Name:=copy(t,62,15);
{ Расстояние между объектами из Tycho-2 и
из каталога спектрального класса }
Val(copy(t,78,6),Dist,err);

{ Звездная величина Mag }
Val(copy(t,85,6),Mag,err);
if Mag>99.9 then Info:=Info or NoMag;
{ Тип звездной величины }
f_Mag:=t[92];

{ Температурный класс, подкласс и класс светимости }
TClass:=t[94];
SClass:=t[95]; if SClass=' ' then Info:=Info or NoS;
LClass:=t[97]; if LClass=' ' then Info:=Info or NoL;

{ Эффективная температура }
Val(copy(t,99,5),Teff,err);
{ Спектральный класс }
SpType:=copy(t,105,20);
end { with };
end { if };
end { ReadTycho2Sp };

end.

```

Листинг 10. Распределение звезд каталога Tycho-2 Spectral Types по классам светимости и спектральным классам

```

Program Tycho2SpDistrib;
{ Распределение звезд Tycho-2 Sp. Type по спектральным классам }
Uses Tycho2Sp;
var star : TTycho2Sp;
const Res = 'spectra.txt'; { Выходной файл }
      NoLum : longint = 0; { Нет данных о классе светимости }
      sp : string[6] = 'BAFGKM'; { Для организации вывода }
var { Общее распределение по классам светимости }
    Lum : array [1..7] of longint;
    { Распред. по спектральным классам и классам светимости }
    TwoDim : array [1..6,1..6] of longint;
    i,j : integer;
    f : text;
begin
for j:=1 to 6 do Lum[j]:=0; { Обнуление данных статистики }
for i:=1 to 6 do for j:=1 to 6 do TwoDim[i,j]:=0;
OpenTycho2Sp;
while ReadTycho2Sp(star) do
with star do begin
if LClass<>' '
then begin { Есть класс светимости }
case TClass of { Разбор температурного класса }
'W','O','B' : i:=1;
'A' : i:=2;
'F' : i:=3;
'G' : i:=4;
'K','R' : i:=5;
'M','N','S','C','P' : i:=6;
end;
j:=ord(LClass)-ord('O'); { Класс светимости }
inc(Lum[j]); { Увеличение статистических счетчиков }
inc(TwoDim[i,j]);
end
else inc(NoLum); { Отсутствует класс светимости }
end;
CloseTycho2Sp;
{ Вывод результатов }
assign(f,Res); rewrite(f);
for i:=1 to 6 do writeln(f,i:1,Lum[i]:7);
writeln(f,'-',NoLum:7);
writeln(f); write(f,' ');
for j:=1 to 6 do write(f,j:8);
writeln(f);
for i:=1 to 6 do
begin
write(f,sp[i]:1,' '); for j:=1 to 6 do write(f,TwoDim[i,j]:8);
writeln(f);
end;
close(f);
end.

```

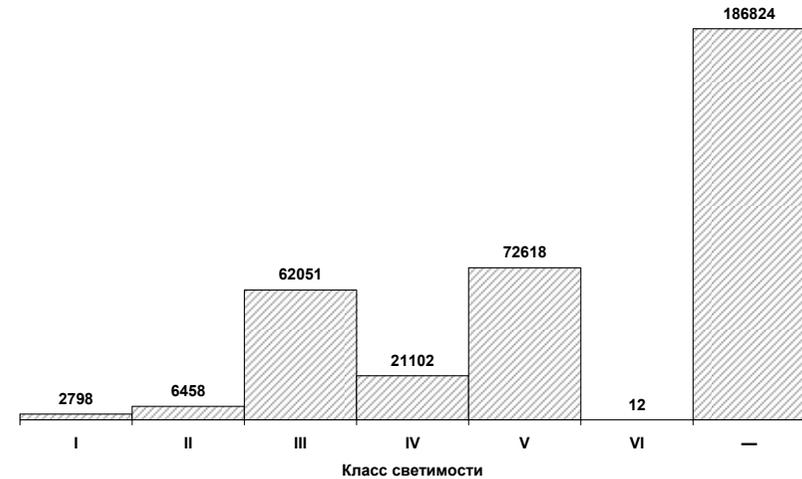


Рис. 16. Распределение звезд каталога Tycho-2 Spectral Types по классу светимости.

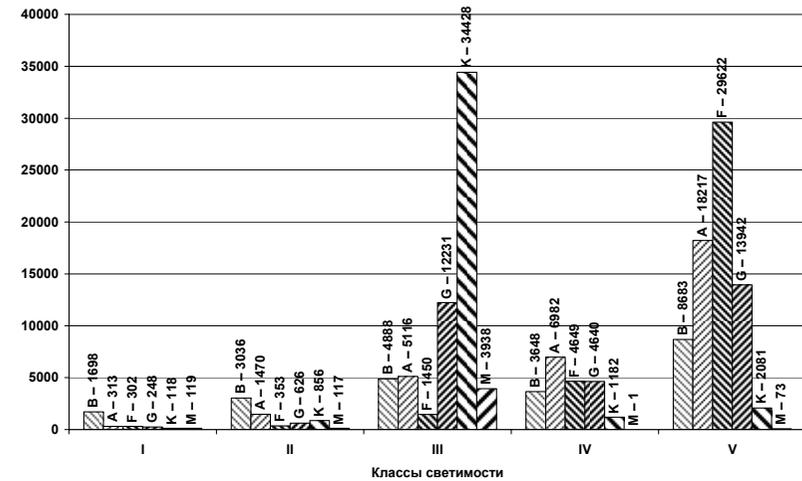


Рис. 17. Распределение звезд каталога Tycho-2 Spectral Types по классу светимости и спектральным классам.

10. Эффективный поиск звезд в каталоге Tycho-2 Spectral Type

В каталоге Tycho-2 Spectral Types нет большинства данных, присутствующих в каталоге Tycho-2, поэтому необходимо решить задачу о быстром поиске спектрального класса для звезд из Tycho-2. Модуль с двумя подпрограммами, решающими эту задачу, представлен на листинге 11. Процедура LoadSpData подготавливает данные для функции

```
FindSpData (T1,T2,T3:word; var TC1,SC1,LC1: char):boolean;
```

Эта функция ищет звезду в каталоге Tycho-2 Spectral Types по своим первым трем параметрам, представляющим идентификаторы звезды в системе Tycho. Если звезда найдена, то функция возвращает значение true, а через последние три параметра поставляет информацию о спектральном классе. Рассмотрим подробнее алгоритм, примененный в этой функции.

Процедура LoadSpData загружает в массив идентификаторы Tycho и спектральную информацию (ее можно без труда расширить, если это будет необходимо для решения конкретной задачи). Параллельно с этим она формирует индексный массив Index, его *i*-й элемент содержит номер первого элемента массива SpData, в котором поле TYC1 равно *i*.

Функция FindSpData с помощью этого массива сразу определяет номер элемента массива SpData, с которого надо начинать поиск. Если внутри цикла окажется, что идентификатор TYC1 уже ссылается на следующую GSC площадку, то из цикла будет совершен досрочный выход, т. е. реальное число переборов звезд будет не более нескольких десятков. В случае совпадения остальных двух идентификаторов звезда считается найденной. Нормальный выход из цикла будет возможен только для звезд в последней 9537-й площадке, в остальных случаях выход осуществляется оператором break.

Используем подготовленный модуль для модификации программы построения распределения звезд по небесной сфере (листинг 6). Построим распределение звезд гигантов (класс светимости III).

Добавим подключение модуля LoadSp, описание дополнительных констант и переменных (листинг 12). Перед чтением каталога Tycho-2 выполним процедуру LoadSpData. В цикле чтения звезд каталога поместим условные операторы, проверяющие наличие звезд в списке Tycho-2 Spectral Types и выполнение условия на желаемый класс светимости.

Листинг 11. Поиск звезды в каталоге Tycho-2 Spectral Types

```
Unit LoadSp;
Interface
Uses Tycho2SP;

Type TSpData=record { Тип элементов массива SpData }
  TYC1 : word; { Идентификатор TYC1 }
  TYC2 : word; { Идентификатор TYC2 }
  TYC3 : word; { Идентификатор TYC3 }
  TClass : char; { Температурный класс }
  SClass : char; { Температурный подкласс }
  LClass : char; { Класс светимости }
end;

Var { Массив соответствий номеров звезд и спектральной информ. }
  SpData : array[1..Tycho2SPNumOfStars] of TSpData;
  { Индексный массив для ускорения поиска }
  Index : array [1..9537] of longint; { GSC площадок - 9537 }

Procedure LoadSpData; { Загрузка данных Tycho-2 Spectral Types }
Function FindSpData { Поиск информации }
  (T1,T2,T3:word; { Номер звезд }
   var TC1, SC1, LC1 : char { Возвращаемая информация }
  ):boolean; { True - звезда найдена }

Implementation

Procedure LoadSpData;
var i : longint;
    star : TTycho2Sp;
    k : longint;
begin
  k:=0;
  OpenTycho2Sp;
  for i:=1 to Tycho2SPNumOfStars do
  with star do begin { Заполнение массива номерами и спектрами }
    ReadTycho2Sp(star);
    SpData[i].TClass:=TClass;
    SpData[i].SClass:=SClass;
    SpData[i].LClass:=LClass;
    SpData[i].TYC1:=TYC1;
    SpData[i].TYC2:=TYC2;
    SpData[i].TYC3:=TYC3;
    if TYC1>k then { Первый номер новой площадки GSC }
    begin
      k:=TYC1; { Элемент массива будет содержать номер первой }
      Index[k]:=i; { записи, соответствующей номеру площадки GSC }
    end;
  end;
end;
CloseTycho2Sp;
end;
```

```

Function FindSpData { Поиск информации }
  (T1,T2,T3:word; { Номер звезд }
   var TC1, SC1, LC1 : char { Возвращаемая информация }
   ):boolean; { True - звезда найдена }
var i : longint;
begin
  FindSpData:=false; { Звезда пока не найдена }
  i:=Index[T1]; { Поиск с записи i, так как в ней первая звезда
                с номером T1 }
  { Поиск до конца массива, чаще выход осуществляется по break }
  while i<Tycho2SPNumOfStars do
    with SpData[i] do
      begin
        if (TYC1>T1) then break; { Началась следующая площадка }
        if (TYC2=T2) and (TYC3=T3) then
          begin { Звезда нашлась }
            TC1:=TClass; SC1:=SClass; LC1:=LClass;
            FindSpData:=true;
            break; { Досрочный выход }
          end;
        inc(i); { Переход к следующему элементу массива }
      end;
    end;
  end.

```

Листинг 12. Изменения в листинге 6

```

Program PictureSp;
Uses CRT, Galactic, Graph, Tycho2, Projection, LoadSp;
Const Lmin='3'; { Минимальный класс светимости }
      Lmax='3'; { Максимальный класс светимости }
      . . . . .
      Grades = 4; { Число градаций серого цвета в изображении }
      . . . . .
Var Sp : array[1..2] of char;
    Lum : char;
. . . . .
OpenTycho2;
LoadSpData;
while ReadTycho2(s) do
begin
  { Проверка по классу светимости }
  if FindSpData(s.TYC1,s.TYC2,s.TYC3,Sp[1],Sp[2],Lum) then
    if (Lum>=Lmin) and (Lum<=Lmax) then
      begin
        { Проекция в экваториальной системе }
        Aitoff(rad(s.RADeg),rad(s.DEDeg),x,y);
        . . . . .

```

Результат работы этой программы для звезд-гигантов демонстрирует рис. 18. Сразу бросается в глаза неоднородность распределения, вызванная плохим знанием классов светимости в северном экваториальном полушарии. В северном экваториальном полушарии есть только несколько отдельных областей с достаточно плотным размещением звезд. Эти факты объясняются свойствами каталогов, вошедших в Tycho-2 Spectral Types.

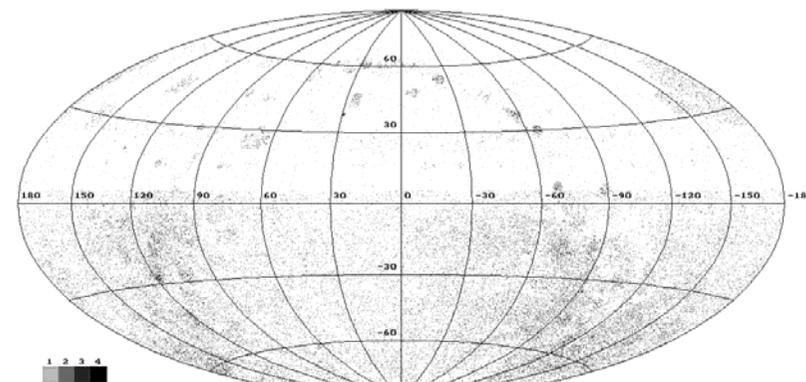


Рис. 18. Распределение звезд класса светимости III по небесной сфере в экваториальной системе координат по данным каталога Tycho-2 Spectral Types. Шкала указывает число звезд на один пиксел изображения.

Приведем еще один пример. Модифицируем программу построения локальной звездной карты (листинг 8) таким образом, чтобы звезды разных спектральных классов отображались различными цветами, а в случае отсутствия информации о спектре использовался бы белый цвет. Эти незначительные изменения отображены в листинге 13.

Основная модификация заключается в вызове функции FindSpData в операторе if и дальнейшей разборке спектрального класса Sp[1] в операторе case.

Обратите внимание, что предложение else color:=White относится к оператору if FindSpData, а не к оператору case.

Листинг 13. Изменения в листинге 8

```

Program LocalChart;
Uses CRT, Galactic, Graph, Tycho2, Projection, Plane, LoadSp;

. . . . .

Var   Sp   : array[1..2] of char;
      Lum  : char;

. . . . .

OpenTycho2;
LoadSpData;

. . . . .

while ReadTycho2(s) do
begin

inc(k); if k mod step = 0 then write('*'); { Бегущая полоса }

{ Проверка по звездной величине }
if (s.magn and NoVT) = 0 then { Существует VT }
if (s.VT>=Mmin) and (s.VT<=Mmax) then { Допустимый диапазон }
{ Проецирование на плоскость }
if ToPlane(A0, D0, s.RAdeg, s.DEdeg, R0, x, y) then
begin
Screen(x, y, CX, CY, scale, u, v); { Экранные координаты }
r:=7-Round(s.VT/2); if r<1 then r:=1;

if FindSpData(s.TYC1, s.TYC2, s.TYC3, Sp[1], Sp[2], Lum) then
{ Задание цвета звезды, если известен спектральный класс }
case Sp[1] of
'W', 'O', 'B' :      color:=LightBlue;
'A' :               color:=LightCyan;
'F' :               color:=LightGray;
'G' :               color:=Yellow;
'K', 'R' :          color:=LightRed;
'M', 'N', 'S', 'C', 'P': color:=Red;
end
else color:=White { Класс не известен }

. . . . .

```

11. Определение спектральных параллаксов звезд каталога Tycho-2 Spectral Types

По сравнению с каталогом Hipparcos большим недостатком каталога Tycho-2 является отсутствие в нем параллаксов звезд. Тем не менее, с помощью данных из каталога Tycho-2 Spectral Types можно получить оценку параллакса звезды. Действительно, зная абсолютную и спектральный класс звезды, можно определить ее абсолютную звездную величину. Сопоставив ее с видимой звездной величиной, можно сделать оценку расстояния. Сложность заключается в учете межзвездного поглощения. Кроме этого, в каталогах спектральных классов звезд далеко не всегда указывается класс светимости, каталог Tycho-2 Spectral Types – тоже не исключение.

Тем не менее, при наличии данных о спектре, показателе цвета и видимой звездной величине можно вычислить параллакс или расстояние с помощью следующего соотношения [27]:

$$\lg \pi = -\lg r = 0.2(M_{\text{табл}} - m_{\text{кат}}) - 1 + 0.2A, \quad (15)$$

где A – вклад межзвездного поглощения, который может быть оценен по формуле

$$A = 3((B - V)_{\text{кат}} - (B - V)_{\text{табл}}). \quad (16)$$

Здесь $m_{\text{кат}}$ и $(B - V)_{\text{кат}}$ – визуальная звездная величина и показатель цвета звезды в каталоге, $M_{\text{табл}}$ и $(B - V)_{\text{табл}}$ – табличная абсолютная звездная величина и показатель цвета для данного спектрального класса, полученные на основе астрофизических исследований. Последние величины можно взять из табл. 7–9 [28]. Данные этих таблиц проиллюстрированы рис. 19. Параллакс звезды, вычисленный по формуле (15), обычно называют спектральным параллаксом. Следуя этой терминологии, величину $r = 1/\pi$ можно назвать «спектральным расстоянием», однако в специальной литературе такой термин отсутствует. По этой причине мы будем называть «спектральное расстояние» просто расстоянием.

Напишем программу (листинг 15), которая строит распределение звезд III и V классов светимости из каталога Tycho-2 Spectral Types по расстоянию. Для вычисления расстояний была написана функция PhotoDistance (листинг 14), использовавшая линейную интерполяцию данных табл. 7 и 8.

Обработанные в электронной таблице результаты показаны на рис. 20. Мы видим, что максимум числа звезд главной последовательности приходится на диапазон 100-200 пк, а красных гигантов – на 300-400 пк. При этом 90 % всех использованных звезд расположены ближе 600 пк, а на расстоянии менее 1400 пк находятся 99 % звезд III и V классов светимости.

Таблица 7. Абсолютные звездные величины и показатели цвета для звезд главной последовательности класса светимости V

Sp	M_V	$B - V$	Sp	M_V	$B - V$
O5	-5.7	-0.33	F0	2.7	0.30
O8	-4.9	-0.32	F5	3.5	0.44
B0	-4.0	-0.30	G0	4.4	0.58
B3	-1.6	-0.20	G5	5.1	0.68
B5	-1.2	-0.17	K0	5.9	0.81
B8	-0.2	-0.11	K5	7.4	1.15
A0	0.6	-0.02	M0	8.8	1.40
A5	1.9	0.15	M5	12.3	1.64

Таблица 8. Абсолютные звездные величины и показатели цвета для красных гигантов класса светимости III

Sp	M_V	$B - V$
G5	0.9	0.86
K0	0.7	1.00
K5	-0.2	1.50
M0	-0.4	1.56
M2	-0.4	1.59
M5	-0.6	1.63

Таблица 9. Абсолютные звездные величины и показатели цвета для сверхгигантов класса светимости Ia, Ia0, Ib

SP	Ib		Ia		Ia0
	M_V	$B - V$	M_V	$B - V$	M_V
O5		-0.32	-6.8	-0.31	
B0	-6.1	-0.24	-6.9	-0.23	-8.2
B5	-5.4	-0.10	-7.0	-0.08	-8.4
A0	-5.2	-0.01	-7.1	0.02	-8.5
A5	-5.1	0.09	-7.4	0.09	-8.8
F0	-5.1	0.19	-8.0	0.17	-9.0
F5	-5.1	0.33	-8.0	0.31	-9.0
G0	-5.0	0.76	-8.0	0.75	-8.9
G5	-4.6	1.00	-7.9	1.03	-8.6
K0	-4.3	1.20	-7.7	1.25	-8.5
K5	-4.4	1.59	-7.5	1.60	
M0	-4.5	1.64	-7.0	1.67	-8.0

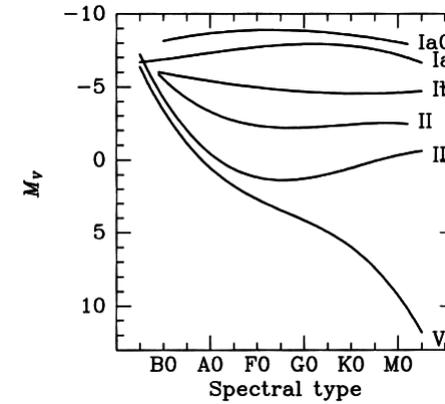


Рис. 19. Зависимость светимости от спектрального класса.

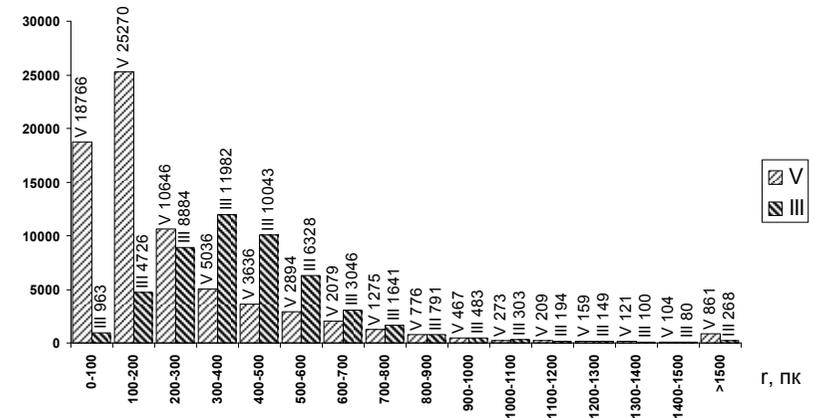


Рис. 20. Распределение звезд III и V класса светимостей каталога Tycho-2 Spectral Types по фотометрическим расстояниям.

Рассмотрим подробнее листинг 14. В качестве параметров функции здесь задаются температурный класс и подкласс, класс светимости, видимая звездная величина и показатель цвета $B - V$. Функция возвращает True, в случае успешного вычисления расстояния, которое передается через последний изменяемый параметр функции.

Данные табл. 7 и 8 с шагом 5 единиц спектрального подкласса содержат массивы Mv_V, BV_V, Mv_III, BV_III.

Первый условный оператор совершает досрочный выход из функции, если класс светимости отличается от III или V. Здесь можно было бы поступить и по-другому: в случае отсутствия класса светимости предположить класс светимости V для звезд ранее G5 и III для G5-K9, как это было сделано авторами Tucho-2 Spectral Types при определении эффективных температур.

Оператор case переводит буквенное обозначение спектрального класса в числовой эквивалент. Затем к этому значению добавляется численное значение подкласса, причем если подкласс не определен, то он полагается равным 5. Таким образом, каждому спектральному классу будет соответствовать значение от 5 до 65.

Далее происходит проверка класса светимости и допустимое для него значение спектрального класса. Спектральный класс для каждого класса светимости приводится в интервал с нижней границей 0. После вычисления линейного смещения x , определяется его целая часть k , т. е. номер левого элемента массива в интерполяционной таблице, и дробная часть, которая опять помещается в x .

В последнем операторе case для каждого класса светимости вычисляются методом линейной интерполяции значения Mtab и BVtab, которые соответствуют величинам $M_{табл}$ и $(B - V)_{табл}$ в формулах (15), (16).

В конце тела функции по этим формулам вычисляется расстояние r , выраженное в пк.

Программа, текст которой помещен в листинге 15, строит статистику распределения вычисленных расстояний звезд каталога Tucho-2 Spectral Types. Алгоритм этой программы прост, однако сделаем несколько замечаний. В качестве видимой звездной величины и показателя цвета $B - V$ программа использует формулы (5) и (6) для преобразования фотометрических данных в системе Tucho в шкалу Джонсона. Далее при успешном определении расстояния с помощью функции PhotoDistance увеличивается на единицу соответствующий элемент массива для накопления данных о числе звезд III или V класса светимости.

Листинг 14. Вычисление расстояния методом спектрального параллакса

```
Unit PhotoDist; { Вычисление расстояния }
Interface

Function PhotoDistance(
    TClass : char; { Температурный класс }
    SClass : char; { Температурный подкласс }
    LClass : char; { Класс светимости }
    MagVis : single; { Визуальная звездная величина }
    B_V : single; { Показатель цвета звезды }
    Var r : single { Результат - расстояние в пк }
):boolean; { True - в случае успешного определения }

Implementation

Const
{ Данные табл. 7 }

{ Абсолютная звездная величина звезд класса светимости V }
Mv_V : array [0..12] of single =
(-5.7, -4.0, -1.2, 0.6, 1.9, 2.7, 3.5, 4.4, 5.1, 5.9, 7.4, 8.8, 12.3);
{ O5 B0 B5 A0 A5 F0 F5 G0 G5 K0 K5 M0 M5 }

{ Показатель цвета звезд класса светимости V }
BV_V : array [0..12] of single = (-0.33, -0.30, -0.17,
-0.02, 0.15, 0.30, 0.44, 0.58, 0.68, 0.81, 1.15, 1.40, 1.64);
{ Данные табл. 8 }

{ Абсолютная звездная величина звезд класса светимости III }
Mv_III : array [0..4] of single = (0.9, 0.7, -0.2, -0.4, -0.3);
{ G5 K0 K5 M0 M5 }

{ Показатель цвета звезд класса светимости III }
BV_III : array [0..4] of single = (0.86, 1.00, 1.50, 1.56, 1.63);

Function PhotoDistance(
    TClass : char; { Температурный класс }
    SClass : char; { Температурный подкласс }
    LClass : char; { Класс светимости }
    MagVis : single; { Визуальная звездная величина }
    B_V : single; { Показатель цвета звезды }
    Var r : single { Результат - расстояние в пк }
):boolean; { True - в случае успешного определения }

var SubCl, SpCl, Sp : integer; { Для числ. представления Sp }
x : single; { Для интерполяции }
k : integer; { Для интерполяции }
Mtab, BVtab : single; { Табличные значения Mv и B-V }
```

```

begin
  PhotoDistance:=False; { Пока не расстояние не определено }
  if (LClass<>'3') and (LClass<>'5') then exit; { III и V }
  case TClass of { Численное представление TClass }
    'W','O' : SpCl:=0;
    'B' : SpCl:=10;
    'A' : SpCl:=20;
    'F' : SpCl:=30;
    'G' : SpCl:=40;
    'K','R' : SpCl:=50;
    'M','N','S','C','P': SpCl:=60;
  else exit;
end;
if SClass=' ' then SClass:='5'; {Если не указан SClass, то 5}
SubCl:=Ord(SClass)-Ord('0'); {Численное представление SClass}
Sp:=SpCl+SubCl; { Полное численное представление сп. класса }
Case LClass of { Проверка допустимого значения Sp }
  '5' : begin
    if (Sp<5) or (Sp>65) then exit;
    Sp:=Sp-5; { от 0 до 60 }
  end;
  '3' : begin
    if (Sp<45) or (Sp>65) then exit;
    Sp:=Sp-45; { от 0 до 20 }
  end;
else exit;
end;
x:=Sp/5.0; { Шаг Sp в табл. 5 }
k:=trunc(x); x:=frac(x); { Интерполяционные коэффициенты }
Case LClass of
  '5' : if k=12
    then begin
      Mtab:=Mv_V[k]; BVtab:=BV_V[k];
    end
    else begin
      Mtab :=Mv_V[k]+x*(Mv_V[k+1]-Mv_V[k]);
      BVtab:=BV_V[k]+x*(BV_V[k+1]-BV_V[k]);
    end;
  '3' : if k=4
    then begin
      Mtab:=Mv_III[k]; BVtab:=BV_III[k];
    end
    else begin
      Mtab :=Mv_III[k]+x*(Mv_III[k+1]-Mv_III[k]);
      BVtab:=BV_III[k]+x*(BV_III[k+1]-BV_III[k]);
    end;
else exit;
end; { case }
r:=0.2*(MagVis-Mtab)+1-0.2*3.0*(B_V-BVtab); { lg r }
r:=exp(r*ln(10)); { r:=10^r - расстояние в пк }
PhotoDistance:=True;
end;
end.

```

Листинг 15. Распределение звезд Tycho-2 Spectral Type по расстояниям

```

Program PhotometricDistances;

Uses Tycho2Sp,PhotoDist;

const Res = 'photodistances.txt'; { Выходной файл }
      N = 15; { Размер массивов статистики }
      delta = 100.0; { Шаг по расстоянию в пк }

var
  { Массивы статистики распределения звезд по расстоянию }
  Dist_V : array [0..N] of longint; { Главная последоват. V }
  Dist_III : array [0..N] of longint; { Гиганты III }

  Mj,BVj : single; { Звездная величина и B-V в системе Джонсона }
  r : single; { Расстояние в пк }
  i : integer; { Индекс элементов массивов }
  f : text; { Выходной файл }
  star : TTycho2Sp; { Звезда из Tycho-2 Spectral Types }

begin
  for i:=0 to N do
    begin { Обнуление массивов }
      Dist_V[i]:=0; Dist_III[i]:=0;
    end;
  OpenTycho2Sp;
  while ReadTycho2Sp(star) do
    with star do begin
      if ((info and NoVT)<>0) or ((info and NoBT)<>0) then continue;
      Mj :=VT-0.090*(BT-VT);
      BVj:=0.850*(BT-VT);
      if PhotoDistance(TClass,SClass,LClass,Mj,BVj,r) then
        begin
          i:=trunc(r/Delta); { Определение элемента массива }
          { В последний элемент попадают все звезды с r>N*delta }
          if i>N then i:=N;
          case LClass of
            '3' : inc(Dist_III[i]);
            '5' : inc(Dist_V[i]);
          end;
        end;
      end; { With & While }
    CloseTycho2Sp;

    { Вывод результатов }
    assign(f,Res); rewrite(f);
    for i:=0 to N do writeln(f,i*Delta:5:0,'-',(i+1)*Delta:5:0,
      Dist_V[i]:7,Dist_III[i]:7);

  close(f);
end.

```

12. Диаграмма Герцшпрунга-Рессела для звезд каталога Tycho-2 Spectral Types

Возможность оценить расстояния до некоторых звезд Tycho-2 позволяет определить их абсолютные звездные величины и, следовательно, построить диаграмму «Спектр – Абсолютная звездная величина» или «Показатель цвета – Абсолютная звездная величина».

Напишем программу, строящую первый вид диаграммы. При этом разным цветом выделим на ней звезды, принадлежащие разным классам светимости. Такая диаграмма, на которой присутствуют звезды главной последовательности (V класс светимости), красные гиганты (III класс светимости) и сверхгиганты (I класс светимости), изображена на рис. 21.

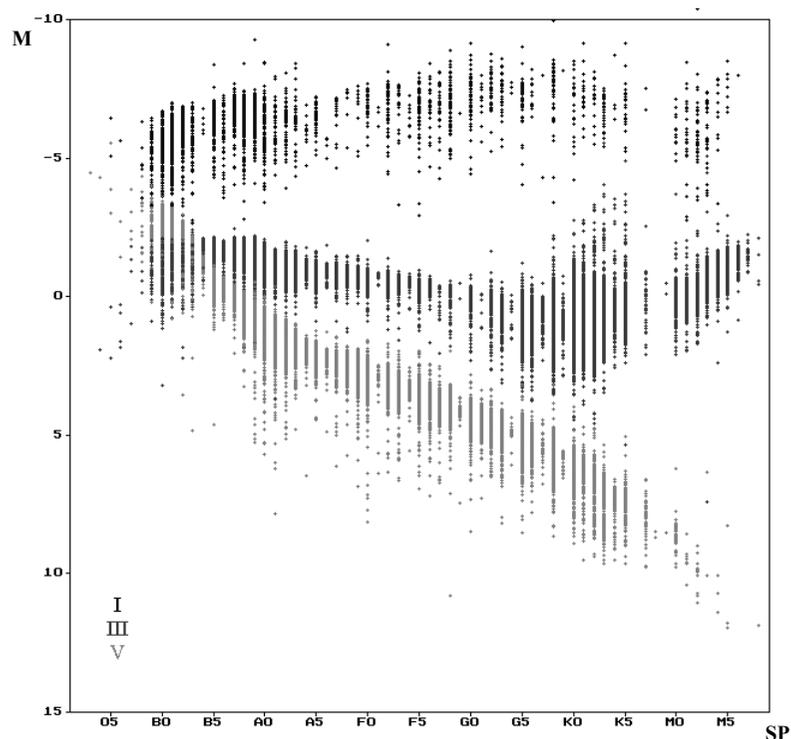


Рис. 21. Диаграмма «Спектр – Абсолютная звездная величина» для звезд I, III и V класса светимостей каталога Tycho-2 Spectral Types.

Мы видим значительную размытость ветвей диаграммы, которая большей частью связана с неточным определением расстояния и спектральной классификации. Но, тем не менее, ветви прослеживаются достаточно отчетливо, это служит указанием на то, что при массовых исследованиях мы можем опираться на расстояния, полученные методом спектральных параллакс.

Текст программы, строящей рис. 21, приведен на листинге 16. В его начале определяются константы и две глобальные переменные, хранящие масштаб по горизонтальной и вертикальной оси. Функция LineSp переводит спектральный класс, заданный в виде двух символов, в число от 0 до 69. Процедура XY, используя предыдущую функцию, вычисляет положение точки на диаграмме для звезды с известным спектральным классом и абсолютной звездной величиной.

Главная программа начинает работу с установления графического окна, вычисления масштабов, построения и разметки осей. Для этого также используется процедура XY.

Тело ключевого в программе цикла while содержит очень простой алгоритм. Вычисляются фотометрические параметры по шкале Джонсона. Функция PhotoDistance дает расстояние r , по которому для каждой звезды индивидуально вычисляется абсолютная звездная величина Mag. После этого можно найти координаты точки на диаграмме. В зависимости от класса светимости ставится точка того или иного цвета.

Для построения диаграммы «Показатель цвета – Абсолютная звездная величина» вместо спектрального класса следует воспользоваться показателем цвета $B-V_j$. Однако для далеких звезд возможно значительное покраснение их цвета, что приведет к изменению $B-V$ и, следовательно, к искажению положения звезд на диаграмме (к смещению вправо). Поэтому такой способ можно применять только для достаточно близких звезд, находящихся на расстоянии не более 100-200 пк.

Листинг 16. Построение диаграммы Герципрунга–Рессела

```

Program HRDiagramm; { Построение HR-диаграммы }
Uses Tycho2Sp, PhotoDist, Math, Graph, CRT;
const
  MagLow =15.0; MagHi =-10.0; { Диапазон по AbsMag }
  Border = 50; { Отступ диаграммы от края окна }
  sps = 'OBAFGKM'; { Для разметки оси }

var { Глобальные переменные }
  ScaleMg : double; { Масштаб по оси абсолют. звездных величин }
  ScaleSP : double; { Масштаб по оси спектров }

{ Перевод спектра в числовую форму }
Function LineSp(Sp,SubSp:char) : integer;
var SpCl:integer;
begin
  case Sp of
    'W','O' : SpCl:=0;
    'B' : SpCl:=10;
    'A' : SpCl:=20;
    'F' : SpCl:=30;
    'G' : SpCl:=40;
    'K','R' : SpCl:=50;
    'M','N','S','C','P': SpCl:=60;
  else SpCl:=-9999;
  end;
  if SubSp=' ' then SubSp:='5';
  LineSp:=SpCl+Ord(SubSp)-Ord('0');;
end;

{ Пересчет спектра и абс. зв. величины в экранные координаты }
Procedure XY(Sp,SubSp:char; Mag:single; var x,y : integer);
begin
  x:=Border+Round(LineSp(Sp,SubSp)*ScaleSP);
  y:=Border+Round((Mag-MagHi)*ScaleMg);
end;

var
  driver, mode : smallint; { Для инициализации графики }
  var star : TTycho2Sp; { Звезда из каталога }
  Mj,BVj : single; { Звездная величина и B-V в системе Джонсона }
  r : single; { Расстояние в пк }
  Mag : single; { Абсолютная звездная величина }
  buf : string[15]; { Буфер для формирования текстовых строк }
  x,y,x1,y1 : integer; { Экранные координаты }
  color : integer; { Цвет точки }
  i : integer; { Для цикла for }

begin
  { Инициализация графического окна }
  driver:=detect; mode:=0; InitGraph(driver,mode,'D:\BP\BGI');

```

```

{ Определение масштабов по осям }
ScaleMg:=(GetMaxY-2*Border)/(MagLow-MagHi);
ScaleSP:=(GetMaxY-2*Border)/(LineSp('M','9')-LineSp('O','1'));

{ Заливка фона и установка цвета }
SetFillStyle(SolidFill,White); Bar(0,0,GetMaxX,GetMaxY);
SetColor(Black);

{ Отрисовка обрамляющей рамки }
XY('O','1',MagHi,x,y);XY('M','9',MagLow,x1,y1);
Rectangle(x,y,x1,y1);

{ Разметка оси спектров }
SetTextJustify(CenterText,TopText);
for i:=1 to Length(Sps) do
begin
  if i<>1 then
  begin
    XY(Sps[i],'0',15,x,y); Line(x,y,x,y+3);
    OutTextXY(x,y+5,Sps[i]+'0');
  end;
  XY(Sps[i],'5',15,x,y); Line(x,y,x,y+3);
  OutTextXY(x,y+5,Sps[i]+'5');
end;

{ Разметка оси абсолютных звездных величин }
SetTextJustify(RightText,CenterText);
Mag:=MagHi;
Repeat
  XY('O','1',Mag,x,y); Line(x,y,x-3,y);
  Str(Mag:0:0,buf); OutTextXY(x-5,y,buf);
  Mag:=Mag+5;
until Mag>MagLow;

OpenTycho2SP;
while ReadTycho2Sp(star) do
with star do begin
  if ((info and NoVT)<>0) or ((info and NoBT)<>0) then continue;
  Mj :=VT-0.090*(BT-VT); BVj:=0.850*(BT-VT); { Джонсон. фотом. }
  if not PhotoDistance(TClass,SClass,LClass,Mj,BVj,r)
  then continue;
  Mag:=Mj-5.0*log10(r)+5.0; { Вычисл. абс. звездной величины }
  XY(TClass,SClass,Mag,x,y);
  case LClass of
    '1' : color:=Blue;
    '3' : color:=Red;
    '5' : color:=Green;
  end;
  PutPixel(x,y,Color);
end;
CloseTycho2Sp;
CloseGraph;
end.

```

13. Совместное использование данных каталогов Hipparcos и Tycho-2

Вопрос о совместном использовании данных для звезд из нескольких каталогов очень часто встает перед астрометристами. Это могут быть задачи сравнения каталогов, например, собственных движений и координат звезд, либо добавления данных из одного каталога к другому. Одна из больших сложностей, возникающих при их решении, – это кросс-идентификация звезд. Решение этой проблемы значительно облегчается, если в обоих каталогах присутствуют идентификаторы звезд в какой-либо общепринятой системе. Обычно это номера в распространенных каталогах: раньше это были BD [29], HD [30] и др. В последнее время в качестве таких систем стали использоваться каталоги Hipparcos и Tycho.

Если же идентифицировать звезды таким образом не удастся, то задача значительно усложняется. В этом случае сравнение каталогов приходится проводить по координатам звезд, переведенных на некоторую общую эпоху. Ошибки в определении координат и собственных движений звезд (особенно при большой разнице эпох) затрудняют выявление общих звезд каталогов. Обычно приходится задаваться некоторым фиксированным угловым расстоянием и считать звезды совпадающими, если после перевода данных обоих каталогов на общую эпоху разность их положений меньше этой величины. В настоящем пособии этот случай мы рассматривать не будем, а ограничимся идентификацией по номерам.

Поставим такую задачу: сравнить расстояния звезд каталога Tycho-2, полученные по спектральным параллаксам, с расстояниями, вычисленными на основе тригонометрических параллаксов звезд каталога Hipparcos. Естественно, такое сравнение можно проводить только по тем звездам каталога Tycho-2 Spectral Types, которые присутствуют в каталоге Hipparcos. Несмотря на кажущуюся простоту задачи, при ее решении возникают несколько тонких моментов, которые мы рассмотрим подробнее.

Программа на листинге 17 использует уже созданные нами модули: Tycho2, PhotoDist, LoadSp, а также модуль HipMain, поставляющий подпрограммы работы с каталогом Hipparcos, его разработку посвящено пособие «Руководство по практической работе с каталогом Hipparcos» [10], а сам текст модуля можно загрузить с сайта Астрономического института СПбГУ [31].

Перед текстом главной программы размещено несколько подпрограмм. Первая – процедура LoadHipparcos, загружающая необходимые нам данные: номер звезд Hip, тригонометрический параллакс и

его ошибку в глобальный (известный во всех подпрограммах) массив записей (структур) HipCat.

Процедура SortHipparcos производит сортировку этого массива в порядке возрастания номеров Hip. Дело в том, что в каталоге Hipparcos звезды отсортированы не по номерам звезд, а по прямым восхождениям. А номера звезд в каталоге Hipparcos ведут свою историю с каталога Hipparcos Input Catalogue [32]. В силу этого номера звезд в каталоге идут не всегда подряд. Для таких «почти упорядоченных массивов» эффективно использовать шейкер-сортировку¹⁹. Этот алгоритм представляет собой модифицированную сортировку «пузырьком»²⁰. Процедура SortHipparcos использует вспомогательную процедуру InterChange, меняющую два элемента массива местами.

Сама сортировка происходит в цикле repeat ... until, который заканчивается в случае истинности флага²¹ Sorted. Первый цикл for последовательно просматривает слева направо пары соседних элементов и в случае неправильного их расположения меняет их местами с помощью процедуры InterChange, которая сбрасывает флаг Sorted. После одного такого прохода в крайней правой позиции будет находиться элемент с самым большим номером Hip, поэтому после каждого прохода мы уменьшаем на единицу переменную r, которая указывает на последний правый отсортированный элемент.

Второй цикл for производит такую же операцию, но в обратном направлении, перемещая к левому концу массива элементы с самым малым значением Hip.

Если во время проходов не будет сделано ни одной перестановки, можно утверждать, что массив отсортирован. Флаг Sorted в этом случае останется истинным, и цикл закончится. Реально при сортировке данных каталога Hipparcos цикл repeat ... until выполняется всего 27 раз.

Сортировка массива нужна для быстрого поиска элемента в нем. Этот поиск выполняется функцией FindHip, которая по номеру звезды находит номер элемента массива, соответствующий этому номеру, либо нуль, если данный номер не найден в массиве. Используется ал-

¹⁹ От англ. shaker – устройство для взбивания коктейлей, которое поворачивают то в одну, то в другую сторону.

²⁰ Подробнее об алгоритмах сортировки и поиска можно узнать в книге Н. Вирта [33].

²¹ Флагом в программировании называют логическую переменную, одно из значений которой сигнализирует о возникновении какого-либо события.

горитм бинарного поиска, который в случае 100 000 элементов находит нужный не более чем за 17 просмотров²².

Главная программа начинается с загрузки каталога Hipparcos и его сортировки. Затем открывается каталог Tycho-2 и загружается каталог Tycho-2 Spectral Types.

Для вычисления спектральных параллаксов достаточно данных только из каталога Tycho-2 Spectral Types, но в нем нет номеров звезд из каталога Hipparcos, которые есть в Tycho-2. Поэтому в основном цикле while взаимная идентификация звезд будет строиться через каталог Tycho-2.

Разберем подробнее структуру тела цикла while. Вначале с помощью функции FindHip мы ищем звезду в каталоге Hipparcos, если номер Hip отсутствует (s.Hip=0) или звезда не найдена в каталоге Hipparcos, переходим к следующему витку цикла оператором continue.

Далее исследуются свойства тригонометрического параллакса: отбрасываются звезды с неположительным параллаксом, плохой точностью или расположенные дальше RMax. После этого функция FindSpData пытается найти по номерам Tycho спектральные характеристики звезды в данных Tycho-2 Spectral Types. Наконец, вычисляется расстояние уже рассмотренным нами ранее способом.

Последний оператор тела цикла выводит в текстовый файл номер звезды по каталогу Hipparcos и две искомые оценки расстояний.

Графическое представление результатов, полученное в результате обработки выходного файла программы в электронной таблице, представлено на рис. 22. Каждая точка показывает одну звезду, ее x-координата является расстоянием, определенным по тригонометрическому параллаксу, а y-координата – по спектральному параллаксу. Если бы расстояния были определены с хорошей точностью, то все точки выстроились бы вдоль прямой $y = x$, но мы видим значительный разброс, который растет с расстоянием, имеется некоторое завышение значения «спектральных» расстояний над тригонометрическими. Такая картина показывает, что индивидуальные оценки спектральных параллаксов звезд следует использовать с осторожностью.

Из задач на совместное использование данных каталогов Hipparcos и Tycho-2 несомненный интерес представляет сравнение собственных движений звезд этих каталогов, мы предлагаем ее решить в качестве самостоятельного упражнения.

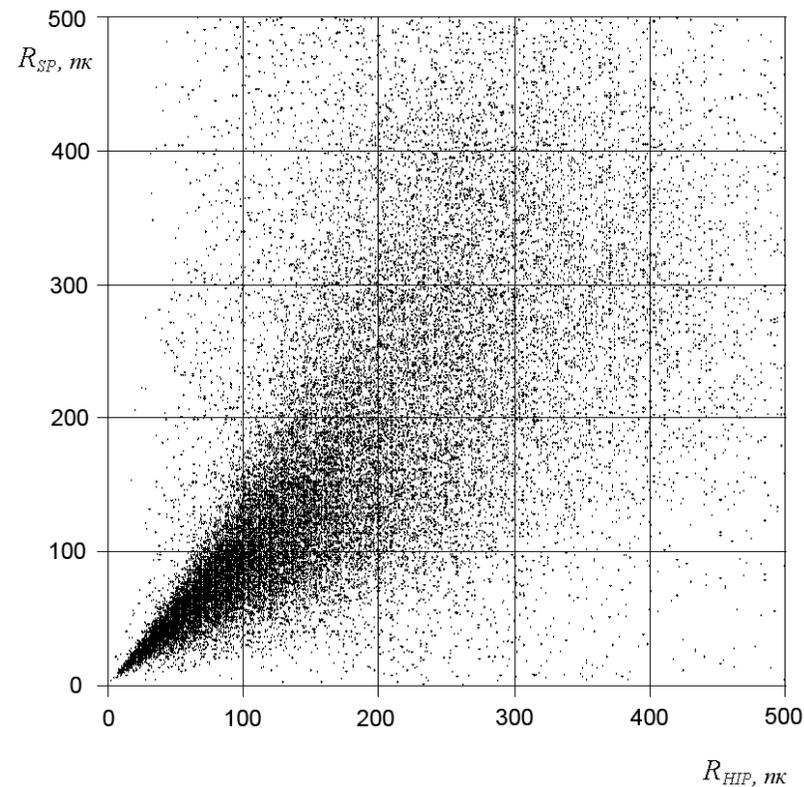


Рис. 22. Диаграмма соответствия между тригонометрическим расстоянием R_{HIP} , полученными по данным каталога Hipparcos, и расстояниями, вычисленными на основе спектральных характеристик звезд каталога Tycho-2 Spectral Types.

²² $2^{17} = 131\,072$, что больше числа звезд в каталоге Hipparcos.

Листинг 17. Сравнение тригонометрических расстояний звезд каталога Hipparcos с расстояниями, полученными на основе каталога Tycho-2 Spectral Types

```

Program HipTycho;
Uses HipMain, Tycho2, PhotoDist, LoadSp;

Const NHip = 117955; { Звезд с параллаксами }
        RelErr = 0.5;  { Допустимая относит. ошибка параллакса }
        Rmax  = 500.0; { Макс. расстояние используемых звезд }

Type THip = record
        Hip  : longint; { Номер звезды в Hipparcos }
        Plx  : single;  { Параллакс звезды в mas }
        SPlx : single;  { Ошибка параллакса в mas }
end;

Var HipCat : array [1..NHip] of THip;

Procedure LoadHipparcos;
{ Загрузка данных каталога Hipparcos в массив HipCat }
var i : longint;
    s : THipparcos;
begin
    OpenHipparcosMain;
    i:=0;
    while ReadHipparcosMain(s) do
    begin
        if (s.info and NoPlx) <> 0 then continue;
        inc(i);
        with HipCat[i] do
        begin
            Hip:=s.Hip; Plx:=s.Plx; SPlx:=s.sigma.Plx;
        end;
    end;
    CloseHipparcosMain;
end;

Procedure SortHipparcos;
{ Пузырьковая сортировка массива HipCat }

var Sorted : boolean; { Флаг - массив отсортирован }

Procedure InterChange(m,n:longint);
{ Меняет местами m-й и n-й элементы HipCat }
var t : THip;
begin
    t:=HipCat[m]; HipCat[m]:=HipCat[n]; HipCat[n]:=t;
    Sorted:=False; { Сброс флага }
end;

```

```

var l, r, i : longint;

begin
    l:=1; r:=NHip;

    repeat
        Sorted:=True; { Установка флага }

        for i:=1 to r-1 do { Проход слева направо }
            if (HipCat[i].Hip>HipCat[i+1].Hip) then InterChange(i,i+1);
            dec(r);

        for i:=r-1 downto 1 do { Проход справа налево }
            if (HipCat[i].Hip>HipCat[i+1].Hip) then InterChange(i,i+1);
            inc(l);

    until Sorted;

end;

Function FindHip(Hip0:longint):longint;
{ Бинарный поиск элемента массива, в котором Hip=Hip0 }
var l, r, n : longint;
begin
    FindHip:=0;
    l:=1; r:=NHip;
    repeat
        n:=(l+r) div 2;
        if HipCat[n].Hip=Hip0 then begin FindHip:=n; break; end;
        if HipCat[n].Hip<Hip0 then l:=n+1 else r:=n;
    until l>=r;
end;

{ Главная программа }

Const step : longint = Tycho2NumOfStars div 80;

var k : longint;
    s : TTycho2;
    h : longint; { Номер элемента массива в Hipparcos }
    Mj,BVj : single; { Звездная вел. и B-V в системе Джонсона }
    TClass,SClass,LClass : char; { Спектральные классы }
    r_px,r_ph : single; { Искомые расстояния в пк }
    f : text; { Файл результатов }

begin
    write('Load Hipparcos ... '); LoadHipparcos; writeln('Ok');
    write('Sort Hipparcos ... '); SortHipparcos; writeln('Ok');
    OpenTycho2;
    write('Load Spectral Data ... '); LoadSpData; writeln('Ok');
    assign(f,'r.txt'); rewrite(f);

```

```

while ReadTycho2(s) do
begin
  { Полоса прогресса из 80 точек }
  inc(k); if k mod step = 0 then write('.');

  { Поиск звезды в Hipparcos }
  if s.Hip=0 then continue else h:=FindHip(s.Hip);
  if h=0 then continue; { Если не найдена, далее пропустить }

  if HipCat[h].Plx<=0.0 then continue; { "Плохой" параллакс }

  { Плохая точность параллакса }
  if HipCat[h].sPlx/HipCat[h].Plx>RelErr then continue;

  r_px:=1000.0/HipCat[h].Plx; { Вычислить тригоном. расстояние }

  if r_px>RMax then continue; { Большое расстояние пропустить }

  { Проверка по звездной величине }
  if ((s.magn and NoVT)<>0) or ((s.magn and NoBT)<>0)
    then continue;

  { Поиск спектральных данных }
  if not FindSpData(s.TYC1,s.TYC2,s.TYC3,TClass,SClass,LClass)
    then continue;

  { Вычисление фотометрического расстояния }
  Mj :=s.VT-0.090*(s.BT-s.VT); BVj:=0.850*(s.BT-s.VT);
  if not PhotoDistance(TClass,SClass,LClass,Mj,BVj,r_ph)
    then continue;

  { Вывод результата }
  writeln(f,HipCat[h].Hip:6,r_px:8:1,r_ph:8:1);

end; { while }

close(f);

end.

```

14. Упражнения

- Проведите статистическое исследование каталога Tycho-2:
 - Постройте распределение звезд каталога по полной ошибке собственного движения $\sigma_{\mu} = \sqrt{\sigma_{\mu_{\alpha^*}}^2 + \sigma_{\mu_{\delta}}^2}$.
 - Постройте распределение звезд каталога по относительной ошибке собственного движения $\mu_{rel} = \frac{\sigma_{\mu}}{\mu}$, здесь $\mu = \sqrt{\mu_{\alpha^*}^2 + \mu_{\delta}^2}$.
 - Узнайте число звезд, для которых средние положения даны для фотоцентра.
 - Постройте гистограмму распределения поля Num – числа использованных положений для определения собственных движений.
 - Постройте распределение звезд каталога по показателю цвета $B - V$, вычисленного по формуле (6), определите самую «голубую» и самую «красную» звезду.
- Используя поле HIP, постройте каталог разностей для собственных движений Tycho-2 и Hipparcos.
 - Постройте распределение разностей модулей собственных движений. Получите математическое ожидание и среднеквадратичное отклонение разности.
 - Определите звезды с аномально большими разностями собственных движений. Проверьте, не является ли эта звезда двойной в Tycho-2 (установлены соответствующие флаги), проанализируйте значение поля proх.
- Постройте распределение звезд каталога Tycho-2 на небесной сфере:
 - для звезд разных величин, например для четырех групп: $m < 7$, $7 \leq m < 9$, $9 \leq m < 11$, $m \geq 11$;
 - для звезд с разными показателями цвета, например для пяти групп: $B - V < -0.5$, $-0.5 \leq B - V < 0.0$, $0.0 \leq B - V < +0.7$, $+0.7 \leq B - V < +1.2$, $B - V \geq +1.2$;

- c) для двадцати групп звезд, являющимися пересечениями групп упражнений а и b.
 - d) Постройте распределение на небесной сфере звезд I, II, IV и V классов светимости.
4. Проведите поиск звезд классов светимости Ia0, Ia, Ib. Для этого проанализируйте значение поля SpType.
 5. Дополните функцию PhotoDistance возможностью работать с I классом светимости. Постройте распределение этих звезд по расстояниям.
 6. Расширьте возможность функции PhotoDistance таким образом, чтобы в случае отсутствия класса светимости предполагался класс светимости V для звезд ранее G5, III для G5-K9 и I для класса M.
 - a) Постройте распределение звезд по расстоянию с помощью новой версии функции.
 - b) Используя модифицированную функцию PhotoDistance, постройте диаграмму Герцшпрунга–Рессела (Абсолютная звездная величина – Спектральный класс).
 7. Постройте диаграмму Герцшпрунга–Рессела, отложив по оси абсцисс не спектральный класс, а показатель цвета.
 8. Используя программу RedShift 4, найдите координаты кратной звезды θ Ori. С помощью функции ToPlane (листинг 7) постройте локальную карту окрестности этой звезды радиусом 4° . Сравните результат с изображением этой же области, которое строит RedShift.
 9. Постройте локальные карты рассеянных звездных скоплений:
 - a) «Ясли» в созвездии Рака;
 - b) χ и h Персея.
 10. Найдите самые необычные объекты в каталогах Tycho-2 и Tycho-2 Spectral Types, имеющие
 - a) самое большое собственное движение;
 - b) самый большой и самый малый показатель цвета;
 - c) самое большое и самое малое расстояние;
 - d) самую большую и самую малую видимую звездную величину;
 - e) самую большую и самую малую абсолютную звездную величину.

15. Перечень листингов программ²³

1. Модуль чтения каталога Tycho-2	14
2. Построение распределения звезд Tycho-2 по видимой звездной величине	19
3. Процедуры вычисления координат звезд в проекции Хаммера–Айтофа	24
4. Процедура, рисующая координатную сетку	25
5. Перевод экваториальных координат в галактические	27
6. Построение распределения звезд по небесной сфере	27
7. Центральная проекция участка небесной сферы на плоскость	33
8. Построение локальной звездной карты	34
9. Чтение каталога Tycho-2 Spectral Types	47
10. Распределение звезд каталога Tycho-2 Spectral Types по классам светимости и спектральным классам	49
11. Поиск звезды в каталоге Tycho-2 Spectral Types	52
12. Изменения в листинге 6	53
13. Изменения в листинге 8	55
14. Вычисление расстояния методом спектрального параллакса	60
15. Распределение звезд Tycho-2 Spectral Type по расстояниям	62
16. Построение диаграммы Герцшпрунга–Рессела	65
17. Сравнение тригонометрических расстояний звезд каталога Hipparcos с расстояниями, полученными на основе каталога Tycho-2 Spectral Types.	71

²³ Справа приведены номера страниц.

Приложение А

Тексты программ на языке С

Листинг А.1. Модуль чтения каталога Tycho-2

Файл tycho2.h

```
// Интерфейсная часть модуля tycho2
// Расположение полной версии каталога Tycho2
#define Tycho2Name "D:/CATALOGS/TYCHO-2/Data/catalog.dat"
#define Tycho2NumOfStars 2539913

typedef struct
{
    // Идентификаторы
    unsigned TYC1; // [1..9537] Идентификатор
    unsigned TYC2; // [1..12121] Идентификатор
    unsigned TYC3; // [1..3] Идентификатор
    // Средние положения и собственные движения на J2000.0
    char pflag; /* [ PX]
        ' ' - нормальное среднее положение и с.д.
        'P' - среднее положение и с.д. фотоцентра,
        'X' - нет среднего положения и с.д. */
    double mRAdeg, mDEdeg; // Средние RA и Dec на J2000.0
    double pmRA, pmDE; // Средние с.д. на J2000.0
    float e_mRA, e_mDE; // Ошибки RA и DE на среднюю эпоху
    float e_pmRA, e_pmDE; // Ошибки собственных движений
    float merRA, merDE; // Средняя эпоха RA и DE
    unsigned Num; // Число использованных положений
    float g_mRA, g_mDE; // Критерий согл. средних RA и DE для Num>2
    float g_pmRA, g_pmDE; // Крит. согл. собств. движений для Num>2
    float VT, e_VT; // Звездная величина VT и ош. (пров. Info)
    float VT, e_VT; // Звездная величина VT и ош. (пров. Info)
    unsigned prox; // Флаг близости
    char TYC; // Наличие звезды в Tycho-1 'T' или ' '
    long HIP; // Номер звезды по Hipparcos или 0
    char CCDM[4]; // Идентификатор CCDM для звезд Hipparcos
    double RAdeg, DEdeg; // RA и DE полученные в Tycho-2, ICRS
    float epRA, epDE; // Эпоха-1990.0 для RAdeg, DEdeg
    float e_RA, e_DE; // Ошибки RAdeg, DEdeg, основанные на модели
    char postflg; /* Тип решения в Tycho2: ' ' - нормальное,
        'D' - двойная звезда, 'P' - фотоцентр */
    float corr; // Корреляция между RAdeg и DEdeg
    struct
    {
        unsigned NoVT:1; // Нет данных о VT
        unsigned NoVT:1; // Нет данных о VT
    } magn; // Битовые флаги отсутствия VT и VT
} Tycho2;
```

```
// Прототипы функций
void OpenTycho2(); // Открытие файла каталога
void CloseTycho2(); // Закрытие файла каталога

/* Чтение одной строки каталога:
    Результат помещается в s,
    возвращает false при достижении конца файла */
bool ReadTycho2(TTycho2 *star);
```

Файл tycho2.c

```
#include "tycho2.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define Tycho2RecSize 208 // Буфер для строки

FILE * cat; // Файловая переменная каталога

void OpenTycho2()
{
    cat=fopen(Tycho2Name,"r+t");
}

void CloseTycho2()
{
    fclose(cat);
}

// Модификация стандартной функции strncpy, которая добавляет
// \0 к концу строки, в отличие от strncpy
char *strncpyz(char *dest, char *src, int count)
{
    strncpy(dest,src,count); dest[count]='\0';
    return dest;
}

// Возвращает True, если s состоит только из пробелов
int IsBlank(char *s)
{
    while (*s) if (*s++ != ' ') return 0;
    return 1;
}

int ReadTycho2(TTycho2 * star)
{
    char t[Tycho2RecSize]; // Буфер для чтения строки каталога
    char st[16]; // Вспомогательный буфер

    // Если достигнут конец файла, то возвращаем False
    if feof(cat) return 0;
    fgets(t,Tycho2RecSize,cat); // Чтение одной строки
```

```

star->magn.NoVT=0; star->magn.NoBT=0; // Обнуление всех битов

// Определение идентификаторов
star->TYC1=atoi(strncpy(st,t+1-1,4));
star->TYC2=atoi(strncpy(st,t+6-1,5));
star->TYC3=atoi(strncpy(st,t+12-1,1));

// Определение средних положений и с.д.
star->pflag=t[14-1];
if (star->pflag!='X')
{ // Средние положения
star->mRAdeg=atof(strncpy(st,t+16-1,12));
star->mDEdeg=atof(strncpy(st,t+29-1,12));
// Собственные движения
star->pmRA=atof(strncpy(st,t+42-1,7));
star->pmDE=atof(strncpy(st,t+50-1,7));
// Ошибки положений
star->e_mRA=atof(strncpy(st,t+58-1,3));
star->e_mDE=atof(strncpy(st,t+62-1,3));
// Ошибки собственных движений
star->e_pmRA=atof(strncpy(st,t+66-1,4));
star->e_pmDE=atof(strncpy(st,t+71-1,4));
// Средняя эпоха RA и DE
star->mepRA=atof(strncpy(st,t+76-1,7));
star->mepDE=atof(strncpy(st,t+84-1,7));
// Число использованных положений
star->Num=atoi(strncpy(st,t+92-1,2));
// Критерии согласия
if (star->Num>2)
{
star->g_mRA=atof(strncpy(st,t+95-1,3));
star->g_mDE=atof(strncpy(st,t+99-1,3));
star->g_pmRA=atof(strncpy(st,t+103-1,3));
star->g_pmDE=atof(strncpy(st,t+107-1,3));
}
} // if pflag

// Фотометрия
// Звездная величина BT
strncpy(st,t+111-1,6);
if (IsBlank(st)) star->magn.NoBT=1;
else { star->BT=atof(st);
star->e_BT=atof(strncpy(st,t+118-1,5));
}

strncpy(st,t+124-1,6);
if (IsBlank(st)) star->magn.NoVT=1;
else { star->VT=atof(st);
star->e_VT=atof(strncpy(st,t+131-1,5));
}

// Флаг близости
star->prox=atoi(strncpy(st,t+137-1,6));

```

```

// Наличие звезды в Tycho-2
star->TYC=t[141-1];

// Номер звезды по Hipparcos или 0
strncpy(st,t+143-1,6);
if (IsBlank(st)) { star->HIP=0; star->CCDM[0]='\0'; }
else { star->HIP=atoi(st); strncpy(star->CCDM,t+149-1,3); }

// RA, DE, полученные в Tycho-2
star->RAdeg=atof(strncpy(st,t+153-1,12));
star->DEdeg=atof(strncpy(st,t+166-1,12));

// Эпохи RA, DE, полученных в Tycho-2
star->epRA=atof(strncpy(st,t+179-1,4));
star->epDE=atof(strncpy(st,t+184-1,4));

// Ошибки RA, DE, полученных в Tycho-2, основанные на модели
star->e_RA=atof(strncpy(st,t+189-1,5));
star->e_DE=atof(strncpy(st,t+195-1,5));

// Тип решения в Tycho-2
star->postflg=t[201];

// Корреляция между RA и DE
star->corr=atof(strncpy(st,t+203-1,4));
return 1;
}

```

Листинг А.2. Построение распределения звезд Tycho-2 по видимой звездной величине

```

// Распределение звезд Tycho-2 по звездной величине
#include "tycho2.h"
#include <stdio.h>
#include <math.h>

#define Delta 0.5 // Шаг по звездной величине
#define MinV -1.0 // Минимальная звездная величина
#define MaxV 16.0 // Максимальная звездная величина
#define Res "statist.txt" // Выходной файл
#define N int((MaxV-MinV)/0.5)

void main()
{
TTycho2 star;
long k=0;
long step = Tycho2NumOfStars/80;
float V; // Звездная величина
long A[N+1];
int i;
FILE *f;

```

```

for (i=0;i<=N;i++) A[i]=0; // Обнуление статистики
OpenTycho2();
while (ReadTycho2(&star))
{
    if (++k % step == 0) printf("");
    // Пропустить звезду, если нет VT или VT
    if (star.magn.NoBT | star.magn.NoVT) continue;
    V=(star.VT-0.090*(star.BT-star.VT));
    i=floor((V-MinV)/Delta+0.5); // Определение индекса звезды
    // Увеличение соотв. элемента массива
    if ((i>=0) && (i<=N)) A[i]++;
}
CloseTycho2();
// Вывод результатов
f=fopen(Res,"wt");
for (i=0; i<=N; i++)
    fprintf(f,"%6.2f  %7d\n",i*Delta+MinV,A[i]);
fclose(f);
}

```

Листинг А.3. Процедуры перевода координат

```

#include <math.h>

void Aitoff
(double l, double b, // Сферические координаты в радианах
 double &x, double &y) // Декартовы координаты
{
    double s;
    if (l>M_PI) l=1-2*M_PI; // Приведение l в диапазон -PI до +PI
    s=sqrt(1.0+cos(b)*cos(l/2.0)); // Знаменатель формул (7)
    x=-2.0*cos(b)*sin(l/2.0)/s;
    y=sin(b)/s;
}

void Screen
(double x, double y, // Декартовы координаты
 int X0, int Y0, // Экранные координаты начала декарт. системы
 double Scale, // Масштаб - точек экрана на единицу длины
 int &u, int&v ) // Экранные координаты
{
    u=X0+floor(Scale*x+0.5);
    v=Y0-floor(Scale*y+0.5);
}

```

Листинг А.4. Процедура, рисующая координатную сетку

```

double rad(double x) // Перевод градусов в радианы
{
    return x/180.0*M_PI;
}

void AitoffGrid
(int Step, // Шаг сетки в градусах
 int X0, int Y0, // Экранные координаты центра проекции
 double Scale, // Масштаб - число точек на единицу длины
 int Gr ) // Флаг - в градусах или в часах разметка долготы
{
    int i,j; // Переменные циклов for
    double l,b; // Галактические координаты
    double x,y; // Декартовы координаты
    int u,v; // Экранные координаты
    char s[6]; // Строка для подписей
    int h; // Для разметки осей

    // Нанесение сетки меридианов
    i=-180; // Первый меридиан -180 градусов

    do { // Цикл по меридианам

        l=rad(i); // Перевод в радианы
        j=-90; // Первая точка меридиана

        do { // Цикл построения вдоль меридиана

            // Вычисление точки меридиана
            b=rad(j); // Перевод в радианы широты
            Aitoff(l,b,x,y); // Перевод в декартовы координаты
            Screen(x,y,X0,Y0,Scale,u,v); // Перевод в экранные коорд.

            // Если точка первая (j=-90), то помещаем графический курсор
            // в точку (u,v) функцией MoveTo, если точка не первая, то
            // "прочерчиваем" курсором линию из предыдущей точки
            // в точку (u,v) функцией LineTo.
            if (j==-90) moveto(u,v); else lineto(u,v);
            j+=5; // Шаг 5 градусов обеспечивают гладкий вид меридиана
        } while (j<=90);

        i+=Step; // Переход к следующему меридиану
    } while (i<=180);
}

```

```

// Нанесение сетки параллелей - аналогично предыдущему
j=-90;
do { // Цикл по параллелям
  b=rad(j);
  i=-180;
  do { // Цикл построения вдоль параллели
    l=rad(i);
    Aitoff(l,b,x,y);
    Screen(x,y,X0,Y0,Scale,u,v);
    if (i==180) moveto(u,v); else lineto(u,v);
    i=i+5;
  } while (i<=180);
  j+=Step;
} while (j<=90);
// Задание свойств шрифта (может зависеть от граф.библ.)
settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
settextjustify(LEFT_TEXT,BOTTOM_TEXT);

// Подписи меридианов вдоль экватора
i=-180;
do{
  // Вычисление координаты точки вывода надписи
  l=rad(i);
  Aitoff(l,0,x,y);
  Screen(x,y,X0,Y0,Scale,u,v);

  // Если Gr истина, то разметка в градусах, иначе - в часах
  if (Gr) h=i; else { h=i/15; if (h<0) h+=24; };

  itoa(h,s,10); // Преобразование значения h в текстовую строку
  outtextxy(u+5,v-5,s); // Вывод текстовой строки

  i+=Step; // Переход к следующему меридиану
} while (i<=180);

// Изменение выравнивания текста
settextjustify(RIGHT_TEXT,BOTTOM_TEXT);

// Подписи параллелей вдоль нулевого меридиана - аналогично
j=-90+Step;
do {
  if (j!=0) // Экватор не подписываем
  {
    b=rad(j);
    Aitoff(0,b,x,y);
    Screen(x,y,X0,Y0,Scale,u,v);
    itoa(j,s,10);
    outtextxy(u-5,v-5,s);
  }
  j+=Step;
} while (j<=90-Step);
}

```

Листинг А.5. Перевод экваториальных координат в галактические

```

// Перевод экваториальных координат в галактические
void Galaxy(
  double a, double d, // Экваториальные координаты
  double &l, double &b) // Галактические координаты
{
  double sa,ca,sd,cd;
  double const Leo = 4.936829261; // 282.85948083
  double const L0 = 0.57477039907; // 32.931918056
  double const si = 0.88998807641; // sin 62.871748611
  double const ci = 0.45598379779; // cos 62.871748611

  a=a-Leo;
  sa=sin(a); ca=cos(a);
  sd=sin(d); cd=cos(d);
  b=asin(sd*ci-cd*si*sa);
  l=atan2(sd*si+cd*ci*sa,cd*ca)+L0;
}

```

Листинг А.6. Построение распределения звезд по небесной сфере

```

void main()
{
  TTycho2 s;
  int driver,mode; // Для инициализации графич. режима
  int color; // Цвет точки
  double l, b; // Галактические координаты
  double x, y; // Декартовы координаты
  int u, v; // Экранные координаты
  int CX, CY; // Экранные координаты центра проекции
  int SC; // Масштаб перевода декартовых коорд. в экранные
  int rgb;
  char t[4];

  // Инициализация графического режима,
  // используем простой 16-цветный режим без палитр.
  driver=DETECT; mode=0; initgraph(&driver,&mode,"D:/BC/BGI");

  // Проекция Айтофа имеет X в диапазоне от -2 до 2, и зазор
  SC=getmaxx()/4-10;
  CX=getmaxx()/2; CY=getmaxy()/2;
  setcolor(15);
  AitoffGrid(30,CX,CY,SC,1); // Вывод сетки координат

  OpenTycho2();
  while (ReadTycho2(&s)) // Цикл чтения звезд
  { if (s.magn.NoVT) continue;

```

```

if ((s.VT>=Vmin) && (s.VT<=Vmax))
{
// Перевод экваториальных координат в радианы, а затем
// в галактические координаты
Galaxy(rad(s.RAdeg),rad(s.DEdeg),l,b);
// Вычисление декартовых координат проекции Айтофа
Aitoff(l,b,x,y);
Screen(x,y,CX,CY,SC,u,v); // Перевод в экранные координаты
color=getpixel(u,v);
switch (color)
{
case BLACK: color=DARKGRAY; break;
case DARKGRAY: color=LIGHTGRAY; break;
case LIGHTGRAY: color=WHITE; break;
default: color=WHITE;
}
putpixel(u,v,color); // Поставить точку
}
}; // if и while
CloseTycho2();
getch();
}

```

Листинг А.7. Центральная проекция участка небесной сферы на плоскость

Файл plane.h

```

// Проекция звезды на сферу
bool ToPlane( // Возвращает true в случае успешной проекции
double A0, double D0, // Координаты центра площадки
double a, double d, // Координаты звезды
double maxr, // Масштабный множитель
double &x, double &y); // Декартовы координаты

void Screen
(double x, double y, // Декартовы координаты
int X0, int Y0, // Экранные координаты начала декарт. системы
double Scale, // Масштаб - точек экрана на единицу длины
int &u, int &v); // Экранные координаты

double rad(double x); // Перевод градусов в радианы

```

Файл plane.c

```

#include <math.h>
#include "plane.h"

// Перевод градусов в радианы
double rad(double x) { return x/180.0*M_PI; }

void Screen
(double x, double y, // Декартовы координаты
int X0, int Y0, // Экранные координаты начала декарт. системы
double Scale, // Масштаб - число точек экрана на единицу длины
int &u, int &v ) // Экранные координаты
{
u=X0+floor(Scale*x+0.5);
v=Y0-floor(Scale*y+0.5);
}

bool ToPlane( // Возвращает true в случае успешной проекции
double A0, double D0, // Координаты центра площадки
double a, double d, // Координаты звезды
double maxr, // Масштабный множитель
double &x, double &y) // Декартовы координаты
{
double SinD0, CosD0, sd,cd,cr,sr,tr,sf,cf,r;
A0=rad(A0); D0=rad(D0);
a=rad(a); d=rad(d);
sd=sin(d); cd=cos(d);
SinD0=sin(D0); CosD0=cos(D0);
cr=SinD0*sd+CosD0*cd*cos(A0-a);
if (cr>cos(rad(maxr)))
{
sr=sqrt(1-cr*cr);
tr=sr/cr;
if (sr<0.000001) { sf=0; cf=0; }
else { sf=(cd*sin(A0-a))/sr;
cf=(sd-cr*SinD0)/(sr*CosD0);
};
r=tr;
x=r*sf;
y=r*cf;
return true;
}
else return false;
}

```

Листинг А.8. Построение локальной звездной карты

```
#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#include <graphics.h>
#include <conio.h>
#include "tycho2.h"
#include "plane.h"

void main()
{
    double const A0 = 56.5,
                D0 = 23.97; // Координаты центра поля зрения
    double const R0 = 4.0; // Радиус по X
    double const Mmin=2; // Минимальная звездная величина
    double const Mmax=10; // Максимальная звездная величина

    int const CX=320, CY=240; // Экранные координаты центра
    double const Time = 5000.0; // Интервал в годах для построения
    // штрихов, соответствующих собственным движениям

    TTycho2 s;
    int driver,mode; // Для инициализации графического режима
    int r; // Радиус окружности
    double a1,d1; // Координаты на новую эпоху
    double x, y; // Декартовы координаты
    double scale; // Масштаб
    int u, v; // Экранные координаты
    char t[8]; // Для легенды
    float B_V;
    int color;

    driver=DETECT; mode=0; // Инициализация графического режима
    initgraph(&driver, &mode, "D:/BC/BGI");
    setttextjustify(CENTER_TEXT,CENTER_TEXT);

    // Заливка белым цветом всего окна
    setfillstyle(SOLID_FILL,WHITE); bar(0,0,getmaxx(),getmaxy());
    setcolor(BLACK);

    for (r=2;r<=13;r++)
    {
        circle(r*20,getmaxy()-20,7 - r/2);
        itoa(r,t,10);
        outtextxy(r*20,getmaxy()-30,t);
    }

    scale=(getmaxx()/2)/( sin(rad(R0))/cos(rad(R0)) ); // Масштаб

    OpenTycho2();
```

```
while (ReadTycho2(&s))
{
    // Проверка по звездной величине
    if (s.magn.NoVT || s.magn.NoBT) continue;
    if ((s.VT<Mmin) || (s.VT>Mmax)) continue;
    if (ToPlane(A0,D0,s.RAdeg,s.DEdeg,R0,x,y)
    {
        Screen(x,y,CX,CY,scale,u,v); // Вычисление экранных координат
        r=7-floor(s.VT/2+0.5); if (r<1) r=1;
        B_V=0.850*(s.BT-s.VT);
        if (B_V<-0.5) color=LIGHTBLUE;
        else if (B_V<0.0) color=LIGHTCYAN;
            else if (B_V<0.5) color=YELLOW;
                else if (B_V<1.0) color=LIGHTRED;
                    else color=RED;
        setcolor(BLACK); setfillstyle(SOLID_FILL,color);
        fillellipse(u,v,r,r);

        if (s.pflag!='X')
        {
            a1=s.RAdeg+(s.pmRA*0.001/3600.0)/cos(rad(s.DEdeg))*Time;
            d1=s.DEdeg+(s.pmDE*0.001/3600.0)*Time;
            ToPlane(A0,D0,a1,d1,R0,x,y);
            setcolor(color);
            moveto(u,v);
            Screen(x,y,CX,CY,scale,u,v);
            lineto(u,v);
        }
    }
}
CloseTycho2();
}
```

Листинг А.9. Чтение каталога Tycho-2 Spectral Types

Файл tycho2sp.h

```
// Интерфейсная часть модуля tycho2sp
// Расположение полной версии каталога Tycho2 Spectral Types
#define Tycho2SpName "D:/TYCHO-2 Spectral Types/catalog.dat"
#define Tycho2SpNumOfStars 351863

typedef struct
{
    // Идентификаторы
    unsigned TYC1; // [1..9537] идентификатор
    unsigned TYC2; // [1..12121] идентификатор
    unsigned TYC3; // [1..3] идентификатор
    // Средние положения и собственные движения на J2000.0
    double mRAdeg, mDEdeg; // Средние RA и Dec на J2000.0
    float BT; // Звездная величина BT (пров. Info)
    float VT; // Звездная величина VT (пров. Info)
    char r_SpType[4]; // Источник спектрального класса
    char Name[16]; // Дополнительное обозначение звезды
    float Dist; // Расстояние между объектом Tycho-2 и объектом
                // из каталога спектрального класса
    float Mag; // Зв. величина из каталога спектрального класса
    char f_Mag; // Тип звездной величины [VPBX*]
    char TClass; // Температурный класс
    char SClass; // Температурный подкласс
    char LClass; // Класс светимости
    long Teff; // Эффективная температура
    char SpType[20]; // Развернутое обозначение спектр. класса
    struct
    {
        unsigned NoBT :1; // Нет данных о BT
        unsigned NoVT :1; // Нет данных о VT
        unsigned NoMag:1; // Нет данных о звездной величине Mag
        unsigned NoS :1; // Нет данных о спектральном подклассе
        unsigned NoL :1; // Нет данных о классе светимости
    } info; // Битовые флаги отсутствия фотометрии
} Tycho2Sp;

// Прототипы функций
void OpenTycho2Sp(); // Открытие файла каталога
void CloseTycho2Sp(); // Закрытие файла каталога

/* Чтение одной строки каталога:
   Результат помещается в s,
   возвращает false при достижении конца файла */
bool ReadTycho2Sp(TTycho2Sp *star);
```

Файл tycho2sp.c

```
#include "tycho2sp.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define Tycho2SpRecSize 160 // Длина буфера

FILE * f; // Файловая переменная каталога

void OpenTycho2Sp()
{
    f=fopen(Tycho2SpName,"r+t");
}

void CloseTycho2Sp()
{
    fclose(f);
}

// Замена стандартной функции strncpy,
// strncpyz добавляет \0 к концу строки
char *strncpyz(char *dest, char *src, int count)
{
    strncpy(dest,src,count); dest[count]='\0';
    return dest;
}

// Возвращает true, если s состоит только из пробелов
int IsBlank(char *s)
{
    while (*s) if (*s++ != ' ') return false;
    return true;
}

bool ReadTycho2Sp(TTycho2Sp * star)
{
    char t[Tycho2SpRecSize]; // Буфер для чтения строки каталога
    char st[16]; // Вспомогательный буфер

    // Если достигнут конец файла, то возвращаем False
    if feof(f) return false;
    fgets(t,Tycho2SpRecSize,f); // Чтение одной строки

    star->info.NoVT=0; // Обнуление всех битов флага
    star->info.NoBT=0;
    star->info.NoMag =0; star->info.NoS=0; star->info.NoL=0;

    // Определение идентификаторов
    star->TYC1=atoi(strncpyz(st,t+5-1,4));
    star->TYC2=atoi(strncpyz(st,t+10-1,5));
    star->TYC3=atoi(strncpyz(st,t+16-1,1));
```

```

// Определение средних положений и с.д.
star->mRAdeg=atof(strncpyz(st,t+18-1,12));
star->mDEdeg=atof(strncpyz(st,t+31-1,12));

// Звездная величина VT
star->VT=atof(strncpyz(st,t+44-1,6));
if (star->VT>99.9) star->info.NoVT=1;

// Звездная величина BT
star->BT=atof(strncpyz(st,t+51-1,6));
if (star->BT>99.9) star->info.NoBT=1;

// Источник спектрального класса
strncpyz(star->r_SpType,t+58-1,3);

// Дополнительное имя звезды
strncpyz(star->Name,t+62-1,15);

// Расстояние между объектом Tycho-2 и
// объектом из каталога спектрального класса
star->Dist=atof(strncpyz(st,t+78-1,6));

// Звездная величина Mag
star->Mag=atof(strncpyz(st,t+85-1,6));
if (star->Mag>99.9) star->info.NoMag=1;

// Тип звездной величины
star->f_Mag=t[92-1];

// Температурный класс
star->TClass=t[94-1];

// Температурный подкласс
star->SClass=t[95-1]; if (star->SClass==' ') star->info.NoS=1;

// Класс светимости
star->LClass=t[97-1]; if (star->LClass==' ') star->info.NoL=1;

// Эффективная температура
star->Teff =atoi(strncpyz(st,t+99-1,6));

// Спектральный класс
strncpyz(star->SpType,t+105-1,15);

return true;
}

```

Листинг А.10. Распределение звезд каталога Tycho-2 Spectral Types по классам светимости и спектральным классам

```

#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#include "tycho2sp.h"
#define Res "spectra.txt"

void main()
{
    TTycho2Sp star;
    long NoLum = 0;
    char sp[6] = "BAFGKM"; // Для вывода результатов
    long Lum[6]; // Общее распределение по классам светимости
    long TwoDim[36]; // Распределение по сп. кл. и классам светим.
    int i,j;
    FILE *f;

    for (j=0; j<6; j++) Lum[j]=0; // Обнуление статистики
    for (i=0; i<6; i++) for (j=0; j<6; j++) TwoDim[6*i+j]=0;
    f=fopen(Res,"wt");
    OpenTycho2Sp();
    while (ReadTycho2Sp(&star))
    {
        if (star.LClass!=' ')
        { j=star.LClass-'0'-1;
          switch (star.TClass)
          { case 'W': case 'O': case 'B': i=0; break;
            case 'A': i=1; break;
            case 'F': i=2; break;
            case 'G': i=3; break;
            case 'K': case 'R': i=4; break;
            case 'M': case 'N': case 'S': case 'C': case 'P': i=5;
          }
          Lum[j]++;
          TwoDim[6*i+j]++;
        }
        else NoLum++;
    }
    CloseTycho2Sp();

    // Вывод результатов
    for (i=0; i<6; i++) fprintf(f,"%d %7d\n",i+1,Lum[i]);
    fprintf(f,"- %7d\n\n",NoLum);
    fprintf(f," \n"); fprintf(f,"%8d",j+1); fprintf(f,"\n");
    for (i=0; i<6; i++)
    { fprintf(f,"%c ",sp[i]);
      for(j=0; j<6; j++) fprintf(f,"%8ld",TwoDim[6*i+j]);
      fprintf(f,"\n");
    }
    fclose(f);
}

```

Листинг А.11. Поиск звезды в каталоге Tycho-2 Spectral Types

Файл loadsp.h

```
// Загрузка данных перед первым вызовом FindSpData
void LoadSpData();

bool FindSpData(unsigned T1, unsigned T2, unsigned T3,
                char &TC1, char &SC1, char &LC1);
```

Файл loadsp.c

```
#include "tycho2sp.h"
#include "loadsp.h"

typedef struct
{
    unsigned TYC1; // Идентификатор TYC1
    unsigned TYC2; // Идентификатор TYC2
    unsigned TYC3; // Идентификатор TYC3
    char TClass; // Температурный класс
    char SClass; // Температурный подкласс
    char LClass; // Класс светимости
} TSpData;

TSpData SpData[Tycho2SpNumOfStars];
long Index[9537];

void LoadSpData()
{ long i;
  TTycho2Sp star;
  unsigned long k;

  k=0;
  OpenTycho2Sp();
  for (i=0; i<Tycho2SpNumOfStars; i++)
  {
    ReadTycho2Sp(&star);
    SpData[i].TClass=star.TClass;
    SpData[i].SClass=star.SClass;
    SpData[i].LClass=star.LClass;
    SpData[i].TYC1=star.TYC1;
    SpData[i].TYC2=star.TYC2;
    SpData[i].TYC3=star.TYC3;
    if (star.TYC1>k) // Первый номер новой площадки GSC
    { k=star.TYC1; // Элемент массива будет содержать номер
      Index[k-1]=i; // первой записи, соответ. номеру площадки GSC
    }
  }
}
```

```
bool FindSpData(unsigned T1, unsigned T2, unsigned T3,
                char &TC1, char &SC1, char &LC1)
{
    long i;
    i=Index[T1-1];
    while (i<Tycho2SpNumOfStars)
    {
        if (SpData[i].TYC1>T1) break;
        if ( SpData[i].TYC2==T2 && SpData[i].TYC3==T3)
        {
            TC1=SpData[i].TClass;
            SC1=SpData[i].SClass;
            LC1=SpData[i].LClass;
            return true;
        }
        i++;
    }
    return false;
}
```

Листинг А.12-13. Вычисление среднего собственного движения звезд Tycho-2, имеющих первый класс светимости²⁴

```
#include <math.h>
#include <stdio.h>
#include "loadsp.h"
#include "tycho2sp.h"
#include "tycho2.h"
void main()
{ char Sp[2];
  char Lum ;
  TTycho2 s;
  double mu=0.0;
  long n=0;
  LoadSpData();
  OpenTycho2();
  LoadSpData();
  while (ReadTycho2(&s))
  if (FindSpData(s.TYC1, s.TYC2, s.TYC3, Sp[0], Sp[1], Lum)
      if (Lum=='1') { mu+=sqrt(s.pmRA*s.pmRA+s.pmDE*s.pmDE); n++; }
    mu/=n;
    printf("%4.2lf\n %ld\n",mu,n);
}
```

<p>Вывод программы</p> <p>7.36 2783</p>

²⁴ Использование FindSpData в силу больших размеров массивов требует 32-разрядного компилятора. Так как автору не известен 32-разрядный компилятор с C/C++ для платформы Win32 с BGI-совместимой графической библиотекой, то мы отказались от графического режима в листинге А.12-13 и для простоты изложения использовали пример, который может быть легко откомпилирован в консольную 32-разрядную программу. Читатель может самостоятельно построить программу с использованием графического интерфейса Windows или какой-либо Linux-библиотеки.

Листинг А.14. Вычисление фотометрического расстояния

Файл photodis.h

```
// Вычисление фотометрического расстояния
bool PhotoDistance( // true - в случае успешного определения
    char TClass, // Температурный класс
    char SClass, // Температурный подкласс
    char LClass, // Класс светимости
    float MagVis, // Визуальная звездная величина
    float B_V, // Показатель цвета
    float & r // Результат - расстояние в пк
);
```

Файл photodis.c

```
#include "photodis.h"
#include <math.h>
float const Mv_V[13] =
{ -5.7,-4.0,-1.2,0.6,1.9,2.7,3.5,4.4,5.1,5.9,7.4,8.8,12.3};
// O5 B0 B5 A0 A5 F0 F5 G0 G5 K0 K5 M0 M5
// 0 1 2 3 4 5 6 7 8 9 10 11 12
float const BV_V [13] =
{ -0.33,-0.30,-0.17,-0.02,
// O5 B0 B5 A0
0.15,0.30,0.44,0.58,0.68,0.81,1.15,1.40,1.64};
// A5 F0 F5 G0 G5 K0 K5 M0 M5
float const Mv_III[5] = { 0.9, 0.7, -0.2, -0.4, -0.3 };
// G5 K0 K5 M0 M5
float const BV_III[5] = { 0.86, 1.00, 1.50, 1.56, 1.63 };
// G0 G5 K0 K5 M0 M5
float const Mv_Ia[12] =
{ -6.8,-6.9,-7.0,-7.1,-7.4,-8.0,-8.0,-8.0,-7.9,-7.7,-7.5,-7.0};
// O5 B0 B5 A0 A5 F0 F5 G0 G5 K0 K5 M0
float const BV_Ia[12] =
{-0.31,-0.23,-0.08,0.02,0.09,0.17,0.31,0.75,1.03,1.25,1.60,1.67};

// Вычисление фотометрического расстояния

bool PhotoDistance( // true - в случае успешного определения
    char TClass, // Температурный класс
    char SClass, // Температурный подкласс
    char LClass, // Класс светимости
    float MagVis, // Визуальная звездная величина
    float B_V, // Показатель цвета
    float &r // Результат - расстояние в пк
)
{
    int SubCl, SpCl, Sp;
    float x;
    double y;
    int k;
    float Mtab,BVtab;
    if (LClass!='1' && LClass!='3' && LClass!='5') return false;
```

```
switch(TClass)
{case 'W': case 'O': SpCl=0; break;
 case 'B' : SpCl=10; break;
 case 'A' : SpCl=20; break;
 case 'F' : SpCl=30; break;
 case 'G' : SpCl=40; break;
 case 'K': case 'R': SpCl=50; break;
 case 'M': case 'N': case 'S':
 case 'C': case 'P': SpCl=60; break;
 default: return false;
}
if (SClass==' ') SClass='5';
SubCl=SClass-'0'; Sp=SpCl+SubCl;
switch(LClass)
{case '5': if (Sp<5 || Sp>65) return false;
 Sp=Sp-5;
 break;
 case '3': if (Sp<45 || Sp>65) return false;
 Sp=Sp-45;
 break;
 case '1': if (Sp<5 || Sp>60) return false;
 Sp=Sp-5;
 break;
 default: return false;
}
x=Sp/5.0; x=modf(x,&y); k=int(y);
switch (LClass)
{case '5': if (k==12) { Mtab =Mv_V[k]; BVtab=BV_V[k]; }
 else {
 Mtab =Mv_V[k]+x*(Mv_V[k+1]-Mv_V[k]);
 BVtab=BV_V[k]+x*(BV_V[k+1]-BV_V[k]);
 }
 break;
 case '3': if (k==4) { Mtab =Mv_III[k]; BVtab=BV_III[k]; }
 else {
 Mtab =Mv_III[k]+x*(Mv_III[k+1]-Mv_III[k]);
 BVtab=BV_III[k]+x*(BV_III[k+1]-BV_III[k]);
 }
 break;
 case '1': if (k==11) { Mtab =Mv_Ia[k]; BVtab=BV_Ia[k]; }
 else {
 Mtab =Mv_Ia[k]+x*(Mv_Ia[k+1]-Mv_Ia[k]);
 BVtab=BV_Ia[k]+x*(BV_Ia[k+1]-BV_Ia[k]);
 }
 break;
 default: return false;
}
r=0.2*(MagVis-Mtab)+1.0-0.2*3.0*(B_V-BVtab);
r=pow(10.0,r);
return true;
}
```

Листинг А.15. *Распределение звезд Tycho-2 Spectral Type по фотометрическим расстояниям*

```
#include <stdio.h>
#include <math.h>
#include "tycho2sp.h"
#include "photodis.h"

void main() // Распределение звезд Tycho-2 Spectral Type
{ // по фотометрическим расстояниям
    TTycho2Sp star;

    const char Res[] = "photodis.txt"; // Выходной файл
    long k = 0;
    long step = Tycho2SpNumOfStars / 80;
    int const N = 15; // Размер массивов статистики
    float const delta = 100.0; // Шаг по расстоянию
    // Статистики распределения звезд по расстоянию
    long Dist_V[N+1]; // Главная последовательность
    long Dist_III[N+1]; // Гиганты
    float Mj, BVj; // Звездная величина и B-V в системе Джонсона
    float r; // Расстояние в пк
    int i;
    FILE *f;

    for(i=0; i<=N; i++) { Dist_V[i]=0; Dist_III[i]=0; }
    OpenTycho2Sp();
    while (ReadTycho2Sp(&star))
    {
        if (++k % step == 0) printf("*");
        if (star.info.NoVT || star.info.NoBT) continue;
        Mj =star.VT-0.090*(star.BT-star.VT);
        BVj=0.850*(star.BT-star.VT);
        if (PhotoDistance(star.TClass,star.SClass,star.LClass,Mj,BVj,r))
        {
            i=floor(r/delta);
            if (i>N) i=N;
            switch (star.LClass)
            { case '3' : Dist_III[i]++; break;
              case '5' : Dist_V[i]++; break;
            }
        }
    }
    CloseTycho2Sp();
    // Вывод результатов
    f=fopen(Res,"wt");
    for (i=0; i<=N; i++)
        fprintf(f,"%3.0f-%3.0f%7ld%7ld\n",
            i*delta, (i+1)*delta, Dist_V[i], Dist_III[i]);
    fclose(f);
}
```

Листинг А.16. *Построение диаграммы Герципрунга–Рессела*

```
#include "tycho2sp.h"
#include "photodis.h"
#include <math.h>
#include <graphics.h>
#include <conio.h>
#include <string.h>
#include <stdlib.h>

// Построение HR-диаграммы
const float MagLow =15.0;
const float MagHi =-10.0; // Диапазон по AbsMag
const int Border = 25; // Отступ диаграммы от края окна
char sps[] = "OBAFGKM";
double ScaleMg; // Масштаб по оси абс. зв. величин
double ScaleSP; // Масштаб по оси спектров

int LineSp(char Sp, char SubSp)
{
    int SpCl;
    switch (Sp)
    {
        case 'W': case 'O' : SpCl=0; break;
        case 'B': SpCl=10; break;
        case 'A': SpCl=20; break;
        case 'F': SpCl=30; break;
        case 'G': SpCl=40; break;
        case 'K': case 'R': SpCl=50; break;
        case 'M': case 'N': case 'S':
        case 'C': case 'P': SpCl=60; break;
        default: SpCl=-9999;
    }
    if (SubSp==' ') SubSp='5';
    return SpCl+SubSp-'0';
}

// Процедура пересчета показателя цвета и абс. зв. величины
// в экранные координаты
void XY(char Sp, char SubSp, float Mag, int &x, int &y)
{
    x=Border+floor(LineSp(Sp,SubSp)*ScaleSP+0.5);
    y=Border+floor((Mag-MagHi)*ScaleMg+0.5);
}

void main()
{
    int driver, mode; // Для инициализации графики
    TTycho2Sp star;
    float Mj, BVj; // Звездная величина и B-V в системе Джонсона
    float r; // Расстояние в пк
    float Mag; // Абсолютная звездная величина
    char buf[16]; // Буфер для формирования текстовых строк
```

```

int x,y,x1,y1; // Экранные координаты
int color; // Цвет точки
int i;

// Инициализация графического окна
driver=DETECT; mode=0; initgraph(&driver,&mode,"D:/BC/BGI");

// Определение масштабов по осям
ScaleMg=(getmaxy()-2*Border)/(MagLow-MagHi);
ScaleSP=(getmaxy()-2*Border)/(LineSp('M','9')-LineSp('O','1'));

// Заливка фона и установка цвета
setfillstyle(SOLID_FILL,WHITE); bar(0,0,getmaxx(),getmaxy());
setcolor(BLACK);

// Прорисовка обрамляющей рамки
XY('O','1',MagHi,x,y); XY('M','9',MagLow,x1,y1);
rectangle(x,y,x1,y1);

// Разметка оси спектров
settextjustify(CENTER_TEXT, TOP_TEXT);
buf[2]='\0';
for (i=0; i<strlen(sps); i++)
{
buf[0]=sps[i];
if (i!=0)
{ buf[1]='0';
XY(sps[i],'0',15,x,y); line(x,y,x,y+3);
outtextxy(x,y+5,buf);
}
buf[1]='5';
XY(sps[i],'5',15,x,y); line(x,y,x,y+3); outtextxy(x,y+5,buf);
}

// Разметка оси абс. зв. величин
settextjustify(RIGHT_TEXT,CENTER_TEXT);
Mag=MagHi;
do {
XY('O','1',Mag,x,y); line(x,y,x-3,y);
itoa(int(Mag),buf,10); outtextxy(x-5,y,buf);
Mag+=5;
} while (Mag<=MagLow);

OpenTycho2Sp();

while (ReadTycho2Sp(&star))
{
if (star.info.NoVT || star.info.NoBT) continue;
Mj =star.VT-0.090*(star.BT-star.VT);
BVj=0.850*(star.BT-star.VT);

```

```

if (!PhotoDistance(star.TClass,star.SClass,star.LClass,
Mj,BVj,r)) continue;
Mag=Mj-5.0*log10(r)+5.0; // Вычисл. абс. звезд. величины
XY(star.TClass,star.SClass,Mag,x,y);
switch(star.LClass)
{ case '1' : color=BLUE; break;
case '2' : color=CYAN; break;
case '3' : color=RED; break;
case '4' : color=YELLOW; break;
case '5' : color=GREEN; break;
default: color=DARKGRAY;
};
putpixel(x,y,color);
setcolor(color);
line(x,y-1,x,y+1); line(x-1,y,x+1,y);
} // while

CloseTycho2Sp();
getch();
closegraph();
}

```

Листинг А.17. Сравнение тригонометрических расстояний звезд каталога *Hipparcos* с фотометрическими расстояниями, полученными на основе каталога *Tycho-2 Spectral Types*

```

#include "hipmain.h"
#include "tycho2.h"
#include "photodis.h"
#include "loadsp.h"
#include <stdio.h>

const long NHip = 117955; // Число звезд с параллаксами
const float RelErr = 0.5; // Относит. ошибка параллакса
const float Rmax = 500.0; // Максимальное расстояние

typedef struct
{
long Hip; // Номер звезд в Hipparcos
float Plx; // Параллакс звезды в mas
float sPlx; // Ошибка параллакса в mas
} THip;

THip HipCat[NHip];

```

```

void LoadHipparcos()
{ // Загрузка данных каталога Hipparcos в массив HipCat
  long i;
  THipparcos s;
  OpenHipparcosMain();
  i=0;
  while (ReadHipparcosMain(&s))
  { if (s.Info.NoPlx) continue;
    HipCat[i].Hip=s.HIP;
    HipCat[i].Plx=s.Plx;
    HipCat[i].sPlx=s.sigma.Plx;
    i++;
  }
  CloseHipparcosMain();
} // LoadHipparcos

bool InterChange(long m, long n)
{ // Меняет местами m-й и n-й элементы HipCat
  THip t;
  t=HipCat[m]; HipCat[m]=HipCat[n]; HipCat[n]=t;
  return false;
}

void SortHipparcos()
{ // Пузырьковая сортировка массива HipCat

  bool Sorted; // Флаг - массив отсортирован
  long l, r, i;

  l=0; r=NHip-1;

do
{ Sorted=true; // Установка флага
  for (i=l; i<r; i++) // Проход слева направо
    if (HipCat[i].Hip>HipCat[i+1].Hip) Sorted=InterChange(i,i+1);
  r--;
  for (i=r-1; i>=l; i--) // Проход справа налево
    if (HipCat[i].Hip>HipCat[i+1].Hip) Sorted=InterChange(i,i+1);
  l++;
} while (!Sorted);
} // SortHipparcos

long FindHip(long Hip0)
{ // Бинарный поиск элемента массива, в котором Hip=Hip0
  long l, r, n;
  l=0; r=NHip-1;
  do { n=(l+r)/2;
    if (HipCat[n].Hip==Hip0) return n;
    if (HipCat[n].Hip<Hip0) l=n+1; else r=n;
  } while (l<r);
  return -1;
}

```

```

void main()
{
  const long step = Tycho2NumOfStars / 80;
  long k;
  TTycho2 s;
  long h; // Номер элемента массива в Hipparcos
  float Mj,BVj; // Звездная величина и B-V в системе Джонсона
  char TClass,SClass,LClass; // Спектральные классы
  float r_px,r_ph; // Триг. и фотометрическое расстояния в пк
  FILE *f; // Файл результатов

  printf("Load Hipparcos ... "); LoadHipparcos(); printf("Ok\n");
  printf("Sort Hipparcos ... "); SortHipparcos(); printf("Ok\n");
  OpenTycho2();
  printf("Load Spectral Data ... "); LoadSpData();
  printf("Ok\n");

  f=fopen("r.txt","wt");
  while (ReadTycho2(&s))
  {
    if (k++ % step == 0) printf("*");
    // Поиск звезды в Hipparcos
    h=FindHip(s.HIP);
    if (h<0) continue; // Если звезда не найдена, пропустить
    if (HipCat[h].Plx<=0.0) continue; // "Плохой" параллакс
    // Плохая точность параллакса
    if (HipCat[h].sPlx/HipCat[h].Plx>RelErr) continue;
    r_px=1000.0/HipCat[h].Plx;
    if (r_px>Rmax) continue;
    // Проверка по звездной величине
    if (s.magn.NoVT || s.magn.NoBT) continue;
    // Поиск спектральных данных
    if (!FindSpData(s.TYC1,s.TYC2,s.TYC3,TClass,SClass,LClass))
      continue;

    // Вычисление фотометрического расстояния
    Mj =s.VT-0.090*(s.BT-s.VT); BVj=0.850*(s.BT-s.VT);
    if (!PhotoDistance(TClass,SClass,LClass,Mj,BVj,r_ph))
      continue;
    if (r_ph>Rmax) continue;
    fprintf(f,"%6d%8.1f%8.1f\n",HipCat[h].Hip,r_px,r_ph);
  }
  fclose(f);
}

```

Приложение В

Тексты программ на языке FORTRAN

Листинг В.1. Модуль чтения каталога Tycho-2

```
MODULE Tycho2
IMPLICIT NONE

! Расположение каталога Tycho2
CHARACTER(*), PARAMETER :: Tycho2Name = 'D:\TYCHO-2\catalog.dat'
! Число звезд в Tycho2
INTEGER, PARAMETER :: Tycho2NumOfStars = 2539913
INTEGER, PARAMETER :: u = 11 ! Номер файла

TYPE TTycho2
  INTEGER(2) :: TYC1 ! Идентификатор TYC1 из TYC или GSC
  INTEGER(2) :: TYC2 ! Идентификатор TYC2 из TYC или GSC
  INTEGER(2) :: TYC3 ! Идентификатор TYC3 из TYC

  ! Астрометрические параметры действительны, если pflag!='X'
  CHARACTER(1) :: pflag ! ' ' - нормальное средн. полож. и с.д.
  ! 'P' - среднее полож. и с.д. фотоцентра,
  ! 'X' - нет ни ср. полож., ни с.д., ни их ошибок
  REAL(8) :: mRAdeg, mDEdeg ! (deg) Средние RA и Dec на J2000.0
  REAL(8) :: pmRA, pmDE ! (mas/yr) Средние с.д. на J2000.0
  REAL(4) :: e_mRA, e_mDE ! (mas) Ошибки RA и DE на среднюю эпоху
  REAL(4) :: e_pmRA, e_pmDE ! Ошибки собственных движений
  REAL(4) :: mepRA, mepDE ! Средняя эпоха RA и DE
  INTEGER(4) :: Num ! Число использованных положений
  REAL(4) :: g_mRA, g_mDE ! Критерий согласия RA и DE для Num>2
  REAL(4) :: g_pmRA, g_pmDE ! Критерий согласия с. д. для Num>2

  REAL(4) :: BT, e_BT ! (mag) Звездная вел. BT и ош. (пров. NoBT)
  REAL(4) :: VT, e_VT ! (mag) Звездная вел. VT и ош. (пров. NoVT)
  INTEGER(4) :: prox ! (0.1 arcsec) флаг близости

  CHARACTER(1) TYC ! Наличие звезды в Tycho-1 'T' или ' '
  INTEGER(4) :: HIP ! Номер звезды по Hipparcos или 0
  CHARACTER(3) CCDM ! Идентификатор CCDM для звезд Hipparcos

  REAL(8) :: RAdeg, DEdeg ! (deg) RA и DE получ. в Tycho-2, ICRS
  REAL(4) :: epRA, epDE ! (yr) Эпоха-1990.0 для RAdeg, DEdeg
  REAL(4) :: e_RA, e_DE ! Ошибки RAdeg, DEdeg, основ. на модели

  CHARACTER(1) :: posflg ! Тип решения в Tycho2:
  ! ' ' - нормальное, 'D' - двойная звезда, 'P' - фотоцентр
  REAL(4) :: corr ! Корреляция между RAdeg и DEdeg

  LOGICAL(1):: NoBT ! Нет данных о звездной величине BT
  LOGICAL(1):: NoVT ! Нет данных о звездной величине VT
END TYPE TTYCHO2
```

CONTAINS

```
! Открытие файла каталога в режиме "только чтение"
SUBROUTINE OpenTycho2
  OPEN(u, file = Tycho2Name)
END SUBROUTINE OpenTycho2

! Закрытие файла каталога
SUBROUTINE CloseTycho2
  CLOSE(u)
END SUBROUTINE CloseTycho2

LOGICAL FUNCTION ReadTycho2(s)
TYPE(TTycho2), INTENT(out) :: s
CHARACTER(206) :: buf ! Буфер под строку

100 FORMAT (I4,1X,I5,1X,I1,1X,A1,1X,F12.8,1X,F12.8,1X,F7.1,1X, &
F7.1,1X,I3,1X,I3,1X,F4.1,1X,F4.1,1X,F7.2,1X,F7.2,1X,I2,1X,F3.1, &
1X,F3.1,1X,F3.1,1X,F3.1,1X,F3.1,1X,F6.3,1X,F5.3,1X,F6.3,1X,F5.3,1X,I3, &
1X,A1,1X,I6,A3,1X,F12.8,1X,F12.8,1X,F4.2,1X,F4.2,1X,F5.1,1X, &
F5.1,1X,A1,1X,F4.1)

IF (EOF(u)) THEN
  ReadTycho2=.false.
  RETURN
ELSE
  ReadTycho2=.true.
END IF

read(u, '(A206)') buf
read(buf,100) s.TYC1,s.TYC2,s.TYC3,s.pflag,s.mRAdeg,s.mDEdeg, &
s.pmRA,s.pmDE,s.e_mRA,s.e_mDE,s.e_pmRA,s.e_pmDE,s.mepRA, &
s.mepDE,s.Num,s.g_mRA,s.g_mDE,s.g_pmRA,s.g_pmDE,s.BT, &
s.e_BT,s.VT,s.e_VT,s.prox,s.TYC,s.HIP,s.CCDM,s.RAdeg, &
s.DEdeg,s.epRA,s.epDE,s.e_RA,s.e_DE,s.posflg,s.corr
IF (buf(111:116) .EQ. '') THEN
  s.NoBT=.true.
ELSE
  s.NoBT=.false.
END IF
IF (buf(124:129) .EQ. '') THEN
  s.NoVT=.true.
ELSE
  s.NoVT=.false.
END IF
IF (buf(143:148) .EQ. '') THEN
  s.HIP = 0
  s.CCDM=''
END IF
END FUNCTION ReadTycho2

END MODULE Tycho2
```

Листинг В.2. Построение распределения звезд Тучо-2 по видимой звездной величине

```

PROGRAM MagnDistribution
USE Tycho2

IMPLICIT NONE

TYPE(TTycho2) :: s

REAL(4), PARAMETER :: Delta = 0.5 ! Шаг по звездной величине
REAL(4), PARAMETER :: MinV = -1.0 ! Мин. звездная величина
REAL(4), PARAMETER :: MaxV = +16.0 ! Макс. звездная величина
CHARACTER(*), PARAMETER :: Res = 'statist.txt' ! Выходной файл
INTEGER(4), PARAMETER :: N = FLOOR((MaxV-MinV)/0.5+0.5);
INTEGER(4) :: k = 0
INTEGER(4) :: step = Tycho2NumOfStars/80

REAL(4) :: V ! Звездная величина
INTEGER(4) :: A(0:N) ! Статистика
INTEGER(4) :: i
INTEGER(4), PARAMETER :: f = 20

FORALL (i=0:N) A(i)=0

CALL OpenTycho2

DO WHILE (ReadTycho2(s))
  k=k+1
  IF (mod(k,step)==0) print *, '*'
  IF (s.NoVT .or. s.NoBT) CYCLE ! Нет звездных величин
  V=(s.VT-0.090*(s.BT-s.VT));
  i=FLOOR((V-MinV)/Delta+0.5) ! Определение индекса звезды
  IF ((i>=0) .and. (i<=N)) a(i)=a(i)+1 ! ув.на 1
END DO
CALL CloseTycho2

OPEN(f, file = Res)
WRITE (f, '(F6.2,1X,I7)', (i*Delta+MinV,a(i),i=0,N))
CLOSE(f)

END PROGRAM

```

Листинг В.3. Процедуры перевода координат ²⁵

```

REAL(8), PARAMETER :: PI = 3.1415926535897932384626433832795

SUBROUTINE Aitoff(l,b,x,y)
  REAL(8), INTENT(IN) :: l,b ! Сферические координаты в радианах
  REAL(8), INTENT(OUT) :: x,y ! Декартовы координаты
  REAL(8) :: s, l1 ! Вспомогательные переменные
  IF (l>PI) THEN ! Приведение l в диапазон -Pi до +Pi
    l1=l-2*PI
  ELSE
    l1=l
  END IF
  S=sqrt(1.0+cos(b)*cos(l1/2)) ! Знаменатель формул (7)
  x=-2*cos(b)*sin(l1/2)/s
  y=sin(b)/s
END SUBROUTINE Aitoff

```

Листинг В.4. Процедура, рисующая координатную сетку

```

REAL(8) FUNCTION radi(x) ! Перевод градусов в радианы
  INTEGER, INTENT(IN) :: x
  radi=x/180.0*PI
END FUNCTION radi

REAL(8) FUNCTION rad(x) ! Перевод градусов в радианы
  REAL(8), INTENT(IN) :: x
  rad=x/180.0*PI
END FUNCTION rad

SUBROUTINE AitoffGrid (Step,Gr)

  INTEGER, INTENT(IN) :: Step ! Шаг сетки в градусах
  LOGICAL, INTENT(IN) :: Gr ! Флаг - в градусах или в часах
  INTEGER :: i,j ! Переменные циклов for
  REAL(8) :: l,b ! Галактические координаты
  REAL(8) :: x,y ! Декартовы координаты
  CHARACTER(8) s ! Строка для подписей
  INTEGER :: h ! Для разметки осей

  TYPE (wxycord) wxy
  INTEGER(2) :: status2
  INTEGER(4) :: status4

```

²⁵ Процедура Screen в Digital Fortran не нужна, поскольку в нем можно использовать графическое окно с вещественными координатами с произвольным масштабом и направлением осей.

```

! Нанесение сетки меридианов
status4 = SetColorRGB(#00FF00)

DO i=-180,+180,Step
  l=radi(i) ! Перевод в радианы
  DO j=-90,+90,5 ! Цикл построения вдоль меридиана
    b=radi(j) ! Перевод в радианы широты
    CALL Aitoff(1,b,x,y) ! Перевод в декартовы координаты
    ! Если точка первая (j=-90), то помещаем графический курсор
    ! в точку (x,y) функцией MoveTo_W, если точка не первая, то
    ! "прочерчиваем" курсором линию из предыдущей точки
    ! в точку (u,v) функцией LineTo_W.
    IF (j== -90) THEN
      CALL MoveTo_W(x,y,wxy)
    ELSE
      status2=LineTo_W(x,y);
    END IF
  END DO ! j
END DO ! i

! Нанесение сетки параллелей - аналогично предыдущему
DO j=-90,+90,Step
  b=radi(j)
  DO i=-180,+180,5 ! Цикл построения вдоль параллели
    l=radi(i)
    CALL Aitoff(1,b,x,y);
    IF (i== -180) THEN
      CALL MoveTo_W(x,y,wxy)
    ELSE
      status2=LineTo_W(x,y)
    END IF
  END DO ! i
END DO ! j

status2=SetFont('t'Arial'h10')
status4 = SetColorRGB(#FFFFFF)

! Подписи меридианов вдоль экватора
DO i=-180,+180,Step
  ! Вычисление координаты точки вывода надписи
  l=Radi(i);
  CALL Aitoff(1,0.0_8,x,y)
  ! Если Gr истина, то разметка в градусах, иначе - в часах
  IF (Gr) THEN
    h=i
  ELSE
    h=i/15
    IF (h<0) h=h+24;
  END IF
  write(s,'(I4)') h ! Преобразование значения h в строку
  CALL MoveTo_W(x, y, wxy)
  CALL OUTGTEXT(s)
END DO

```

```

! Подписи параллелей вдоль нулевого меридиана - аналогично
DO j=-90,+90,Step
  if (j /= 0) then ! Экватор не подписываем
    b=Radi(j);
    CALL Aitoff(0.0_8,b,x,y)
    write(s,'(I4)') j
    CALL MoveTo_W(x, y, wxy)
    CALL OUTGTEXT(s)
  end if
END DO

```

END SUBROUTINE AitoffGrid

Листинг В.5. Перевод экваториальных координат в галактические

```

SUBROUTINE Galaxy(a,d,l,b)
REAL(8), INTENT(in) :: a,d
REAL(8), INTENT(out) :: l,b
REAL(8) :: a1,sa,ca,sd,cd

REAL(8), PARAMETER :: Leo = 4.936829261 ! 282.85948083
REAL(8), PARAMETER :: L0 = 0.57477039907 ! 32.931918056
REAL(8), PARAMETER :: si = 0.88998807641 ! sin 62.871748611
REAL(8), PARAMETER :: ci = 0.45598379779 ! cos 62.871748611

a1=a-Leo
sa=sin(a1); ca=cos(a1)
sd=sin(d); cd=cos(d)
b=asin(sd*ci-cd*si*sa)
l=atan2(sd*si+cd*ci*sa,cd*ca)+L0

END SUBROUTINE Galaxy

```

END SUBROUTINE Galaxy

Листинг В.6. Построение распределения звезд по небесной сфере

```

PROGRAM Main

USE DFLIB
USE Tycho2
USE Projection

IMPLICIT NONE

! Физическое разрешение окна
INTEGER, PARAMETER :: MaxX = 1000, MaxY = 500
! Минимальная и максимальная зв. вел.
REAL(8), PARAMETER :: Vmin = 1.0 , Vmax = 15.0

! Логические координаты окна
REAL(8), PARAMETER :: LowX = -2.2 , LowY = -1.05
REAL(8), PARAMETER :: HighX= +2.2 , HighY= +1.05

! Число градаций серого в изображении
INTEGER, PARAMETER :: Grades = 64

REAL(8) :: l, b ! Галактические координаты
REAL(8) :: x, y ! Декартовы координаты

CHARACTER(3) :: t ! Строка для представления числа звезд

LOGICAL status1
INTEGER*2 status2
INTEGER*4 status4
INTEGER*4 color

TYPE(TTycho2) s

TYPE(winDowconfig) wc
TYPE(wxycoord) wxy

wc.numxpixels = MaxX
wc.numypixels = MaxY
wc.numtextcols = -1
wc.numtextrows = -1
wc.numcolors = -1
wc.title = "Aitoff"C
wc.fontsize = #0008000C ! 10 X 12

status1 = SETWINDOWCONFIG(wc)
IF (.NOT. status1) status1 = SETWINDOWCONFIG(wc)

status2=SetWinDow(.TRUE.,LowX, LowY, HighX, HighY)

status2=INITIALIZEFONTS( )
status2=SetFont('t''Arial''h8')
```

```

DO color=0, Grades-1 ! Установка палитры из Grades серых цветов

status4=SETCOLORRGB((65536+256+1)*(color*(256/Grades)))
status4=RECTANGLE_W($GFILLINTERIOR, LowX+0.05*(color-1), &
LowY, LowX+0.05*color, LowY+0.05)

! Преобразование значения color в текстовую строку
write(t,'(I3)') color
status4=SETTEXTCOLORRGB(#FFFFFF)
CALL MoveTo_W(LowX+0.05*(color-0.5)-0.025, LowY+0.10, wxy)
CALL OUTGTEXT(t)
END DO

CALL AitoffGrid(30,.TRUE.)

CALL OpenTycho2 ! Открытие каталога

DO WHILE (ReadTycho2(s)) ! Цикл чтения звезд

IF (s.NoVT/=0) CYCLE
IF ((s.VT<Vmin) .and. (s.VT>Vmax)) CYCLE

! Перевод в галактические координаты
CALL Galaxy(rad(s.RADeg),rad(s.DEdeg),l,b)
CALL Aitoff(l,b,x,y); ! Вычисление декартовых координат

color=GETPIXELRGB_W(x, y)
color=color+(65536+256+1)*(256/Grades)
IF (color>16777215) color=16777215
status4=SetPixelRGB_w(x,y,color)

END DO

CALL CloseTycho2

CALL AitoffGrid(30,.TRUE.)

! Сохранить изображение в файле
status4=SaveImage("Aitoff.bmp"C,0,0,MaxX-1,Maxy-1)

END PROGRAM
```

Листинг В.7. Центральная проекция участка небесной сферы на плоскость²⁶

```
MODULE PLANE
USE DFLIB

IMPLICIT NONE

CONTAINS

Logical Function ToPlane(A0,D0,a,d,maxr,x,y)
! Возвращает .true. в случае успешной проекции
REAL(8),INTENT(IN) :: A0,D0 ! Координаты центра площадки в гр.
REAL(8),INTENT(IN) :: a, d ! Координаты звезды в градусах
REAL(8),INTENT(IN) :: maxr ! Максимальное удаление от центра
REAL(8),INTENT(OUT) :: x,y ! Декартовы координаты

REAL(8) :: SinD0, CosD0, sd,cd,cr,sr,tr,sf,cf,r

sd=dsind(d); cd=dcosd(d)
SinD0=dsind(D0); CosD0=dcosd(D0);
cr=SinD0*sd+CosD0*cd*dcosd(A0-A);
IF (cr>dcosd(maxr)) THEN
  sr=DSQRT(1.0-cr**2);
  tr=sr/cr
  IF (sr<0.0000001) THEN
    sf=0.0; cf=0.0
  ELSE
    sf=(cd*dsind(A0-A))/sr;
    cf=(sd-cr*SinD0)/(sr*CosD0)
  END IF
  r=tr
  X=r*sf; Y=r*cf
  ToPlane=.TRUE.
ELSE
  ToPlane=.FALSE.
END IF
END FUNCTION ToPlane

END MODULE
```

²⁶ В этой функции используются тригонометрические функции от аргумента, выраженного в градусах.

Листинг В.8. Построение локальной звездной карты

```
Program Main

USE DFLIB
USE Tycho2
USE Plane

IMPLICIT NONE

REAL(8),PARAMETER:: A0=56.5,D0=23.97 ! Коорд. центра поля зрения
REAL(8), PARAMETER :: R0 = 4.0 ! Радиус поля зрения
REAL(4), PARAMETER :: Mmin=2 ! Минимальная звездная величина
REAL(4), PARAMETER :: Mmax=10 ! Максимальная звездная величина
REAL(4), PARAMETER :: Time = 10000.0 ! Период в годах для с.д.

INTEGER, PARAMETER :: MaxXY = 700 ! Физическое разрешение окна
REAL(8) :: XYMax ! Логическое разрешение окна
REAL(8) :: scale ! Масштаб
REAL(8) :: x, y ! Декартовы координаты

INTEGER(1) :: status1
INTEGER(2) :: status2
INTEGER(4) :: status4

INTEGER(4) :: color
INTEGER(4) :: r ! Радиус кружка звезды в пикселях
REAL(8) :: rw ! Радиус кружка в оконных координатах
REAL(8) a1,d1 ! Координаты на новую эпоху

INTEGER(2) :: ix,iy, delta ! Для расположения легенды
CHARACTER(2) :: t ! Для подписи легенды

REAL(4) :: B_V ! Вычисляемый показатель цвета

TYPE(TTycho2) s ! Звезда из Tycho-2

TYPE (windowconfig) wc
TYPE (wxycoord) wxy
TYPE (xycoord) xy

wc.numxpixels = MaxXY
wc.numypixels = MaxXY
wc.numtextcols = -1
wc.numtextrows = -1
wc.numcolors = -1
wc.title = "Aitoff"C
wc.fontsize = #0008000C ! 10 X 12

status1 = SETWINDOWCONFIG(wc)
IF (.NOT. status1) status1 = SETWINDOWCONFIG(wc)
```

```

XYmax = DTAND(R0)
scale = (MaxXY/2)/XYmax

status2=SetWindow(.TRUE.,-XYmax, -XYmax, +XYmax, +XYmax)

status2=INITIALIZEFONTS ( )
status2=SetFont('t'Arial'h10')

DO r=2,13 ! Легенда
  ix=R*20; iy=MAXXY-20; delta=7-r/2;
  status4=ELLIPSE($GBORDER,ix-delta,iy-delta,ix+delta,iy+delta)
  write(t,'(I2)') r
  CALL MoveTo(ix-r/2,iy-20,xy)
  CALL OutGText(t)
END DO

CALL OpenTycho2
DO WHILE (ReadTycho2(s))
  ! Проверка по звездной величине
  IF (s.NoVT .or. s.NoBT) CYCLE
  IF (s.VT<Mmin .or. s.VT>Mmax) CYCLE
  IF (ToPlane(A0,D0,s.RAdeg,s.DEdeg,R0,x,y)) THEN
    r=7-FLOOR(s.VT/2+0.5); IF (r<1) r=1
    B_V=0.850*(s.BT-s.VT);
    IF (B_V<-0.5) THEN
      color=RGBTOINTEGER(0,0,255)
    ELSE IF (B_V<0.0) THEN
      color=RGBTOINTEGER(0,127,255)
    ELSE IF (B_V<0.5) THEN
      color=RGBTOINTEGER(255,255,0)
    ELSE IF (B_V<1.0) THEN
      color=RGBTOINTEGER(255,127,0)
    ELSE
      color=RGBTOINTEGER(255,0,0)
    END IF
    status4=SETCOLORRGB(color)
    rw=r/scale
    status4=ELLIPSE_W($GFILLINTERIOR,x-rw,y-rw,x+rw,y+w)
    IF (s.pflag/='X') THEN
      a1=s.RAdeg+(s.pmRA*0.001/3600.0)/dcosd(s.DEdeg)*Time;
      d1=s.DEdeg+(s.pmDE*0.001/3600.0)*Time;
      CALL MoveTo_W(x,y,wxy)
      status4=ToPlane(A0,D0,a1,d1,R0,x,y)
      status4=LineTo_W(x,y)
    ENDIF
  END IF
END DO
CALL CloseTycho2
! Сохранить изображение в файле
status4=SaveImage("LOCALPICTURE.bmp"C,0,0,MaxXY-1,MaxXY-1)
END PROGRAM

```

Листинг В.9. Чтение каталога Tycho-2 Spectral Types

```

MODULE Tycho2Sp

IMPLICIT NONE

! Расположение каталога Tycho2Sp
CHARACTER(*),PARAMETER :: Tycho2SpName='D:\TYCHO2SP\catalog.dat'
! Число звезд в Tycho2 Spectral Types
INTEGER, PARAMETER :: Tycho2SpNumOfStars = 351863

INTEGER, PARAMETER :: usp = 12 ! Номер файла

TYPE TTycho2SP
  ! Идентификаторы
  INTEGER(2) :: TYC1 ! Идентификатор TYC1
  INTEGER(2) :: TYC2 ! Идентификатор TYC2
  INTEGER(2) :: TYC3 ! Идентификатор TYC3
  ! Средние положения на J2000.0
  REAL(8) :: mRAdeg, mDEdeg ! Средние RA и Dec на J2000.0
  REAL(4) :: BT ! Звездная величина BT
  REAL(4) :: VT ! Звездная величина VT
  CHARACTER(3) :: r_SpType ! Источник спектрального класса
  CHARACTER(15) :: Name ! Дополнительное обозначение звезды
  REAL(4) :: Dist ! Расстояние между объектом Tycho-2 и объектом
  ! из каталога спектрального класса
  REAL(4) :: Mag ! Зв. величина из каталога спектрального класса
  CHARACTER(1) :: f_Mag ! Тип звездной величины [VPBX*]
  CHARACTER(1) :: TClass ! Температурный класс
  CHARACTER(1) :: SClass ! Температурный подкласс
  CHARACTER(1) :: LClass ! Класс светимости
  INTEGER(4) :: Teff ! Эффективная температура
  CHARACTER(20) :: SpType ! Развернутое обозначение сп. класса

  LOGICAL(1) :: NoBT ! Нет данных о звездной величине BT
  LOGICAL(1) :: NoVT ! Нет данных о звездной величине VT
  LOGICAL(1) :: NoMag ! Нет данных о звездной величине Mag
  LOGICAL(1) :: NoS ! Нет данных о спектральном подклассе
  LOGICAL(1) :: NoL ! Нет данных о спектральном подклассе
END TYPE TTycho2SP

CONTAINS

! Открытие файла каталога
SUBROUTINE OpenTycho2Sp
  OPEN(usp, file = Tycho2SpName)
END SUBROUTINE OpenTycho2Sp

! Закрытие файла каталога
SUBROUTINE CloseTycho2Sp
  CLOSE(usp)
END SUBROUTINE CloseTycho2Sp

```

```

LOGICAL FUNCTION ReadTycho2Sp(s)
  TYPE (TTycho2Sp), INTENT(out) :: s

  200 FORMAT(4X,I4,1X,I5,1X,I1,1X,F12.8,1X,F12.8,1X,F6.3,1X, &
    F6.3,1X,A3,1X,A15,1X,F6.3,1X,F6.2,1X,A1,1X,A1,A1, &
    1X,A1,1X,I5,1X,A20)

  IF (EOF(usp)) THEN
    ReadTycho2Sp=.false.
    RETURN
  ELSE
    ReadTycho2Sp=.true.
  END IF

  READ(usp,200) s.TYC1,s.TYC2,s.TYC3,s.mRAdeg,s.mDEdeg,s.VT, &
    s.BT,s.r_SpType, s.Name,s.Dist,s.Mag,s.f_Mag,s.TClass, &
    s.SClass, s.LClass,s.Teff,s.SpType

  s.NoVT=s.VT>99.9; s.NoBT=s.BT>99.9; s.NoMag=s.Mag>99.9
  s.NoS=s.SClass==' '; s.NoL=s.LClass==' '

END FUNCTION ReadTycho2Sp

END MODULE Tycho2Sp

```

Листинг В.10. *Распределение звезд каталога Tycho-2 Spectral Types по классам светимости и спектральным классам*

```

PROGRAM MAIN
! Распределение звезд Tycho-2 Spectral Type по спектр. классам
USE Tycho2Sp

IMPLICIT NONE

TYPE (TTycho2Sp) star

CHARACTER(*), PARAMETER :: Res = 'spectra.txt' ! Выходной файл
INTEGER(4) :: NoLum = 0
CHARACTER(6) :: sp = 'BAFGKM'

! Общее распределение по классам светимости
INTEGER(4) :: Lum(1:7)
! Распределение по спектральным классам и классам светимости
INTEGER(4) :: TwoDim(1:6,1:6)

INTEGER(4) :: i,j
INTEGER(4), PARAMETER :: f = 20

```

```

Lum=0; TwoDim=0 ! Обнуление статистики

OPEN(UNIT=f, FILE=Res)
CALL OpenTycho2Sp

DO WHILE (ReadTycho2Sp(star))
  IF (star.LClass/= ' ') THEN
    j=ICHAR(star.LClass)-ICHAR('0')

    SELECT CASE(star.TClass)
      CASE('W','O','B')
        i=1
      CASE('A')
        i=2
      CASE('F')
        i=3
      CASE('G')
        i=4
      CASE('K','R')
        i=5
      CASE('M','N', 'S', 'C', 'P')
        i=6
    END SELECT

    Lum(j)=Lum(j)+1
    TwoDim(i,j)=TwoDim(i,j)+1
  ELSE
    NoLum=NoLum+1
  END IF
END DO

CALL CloseTycho2Sp

! Вывод результатов

WRITE(f,"(I1,I7)") (i,Lum(i),i=1,6)
WRITE(f,"('-',I7)") NoLum
WRITE(f,"(1X)")
WRITE(f,"(' ')")
WRITE(f,"(6I8)") (j,j=1,6)
DO i=1,6
  WRITE(f,"(A1,' ',6I8)") sp(i:i),(TwoDim(i,j),j=1,6);
END DO

CLOSE(f)

END PROGRAM

```

Листинг В.11. Поиск звезды в каталоге Tycho-2 Spectral Types

```
MODULE LoadSp
USE Tycho2SP
IMPLICIT NONE
TYPE TSpData
  INTEGER(2):: TYC1 ! Идентификатор TYC1
  INTEGER(2):: TYC2 ! Идентификатор TYC2
  INTEGER(2):: TYC3 ! Идентификатор TYC3
  CHARACTER(1) :: TClass ! Температурный класс
  CHARACTER(1) :: SClass ! Температурный подкласс
  CHARACTER(1) :: LClass ! Класс светимости
END TYPE
TYPE(TSpData), STATIC :: SpData(1:Tycho2SPNumOfStars)
INTEGER(4), STATIC :: Index(1:9537)
CONTAINS
SUBROUTINE LoadSpData
  INTEGER(4) :: i,k
  LOGICAL :: l
  TYPE(TTycho2Sp) :: star
  k=0
  CALL OpenTycho2Sp
  DO i=1,Tycho2SPNumOfStars
    l=ReadTycho2Sp(star)
    SpData(i).TClass=star.TClass
    SpData(i).SClass=star.SClass
    SpData(i).LClass=star.LClass
    SpData(i).TYC1=star.TYC1
    SpData(i).TYC2=star.TYC2
    SpData(i).TYC3=star.TYC3
    IF (star.TYC1>k) THEN
      k=star.TYC1
      Index(k)=i
    END IF
  END DO
  CALL CloseTycho2Sp
END SUBROUTINE
```

```
LOGICAL FUNCTION FindSpData(T1,T2,T3, TC1,SC1,LC1)
  INTEGER(2),INTENT(IN):: T1,T2,T3
  CHARACTER(1),INTENT(OUT) :: TC1, SC1, LC1
  INTEGER(4) :: i
  FindSpData=.false.
  i=Index(T1);
  DO WHILE (i<Tycho2SPNumOfStars)
    IF (SpData(i).TYC1>T1) EXIT
    IF (SpData(i).TYC2==T2 .and. SpData(i).TYC3==T3) THEN
      TC1=SpData(i).TClass
      SC1=SpData(i).SClass;
      LC1=SpData(i).LClass
      FindSpData=.true.
      EXIT
    END IF
    I=I+1
  END DO
END FUNCTION FindSpData
END MODULE
```

Листинг В.12. Изменения в листинге В.6

```
Program Main
. . . . .
USE LoadSp
. . . . .
CHARACTER(1) :: Sp(2), Lum
. . . . .
CALL OpenTycho2 ! Открытие каталога
CALL LoadSpData
  DO while (ReadTycho2(s)) ! Цикл чтения звезд
    IF (s.NoVT/=0) CYCLE
    IF ((s.VT<Vmin) .and. (s.VT>Vmax)) CYCLE
    IF (.not. FindSpData(s.TYC1,s.TYC2,s.TYC3,Sp(1),Sp(2),Lum)) &
      CYCLE
    IF (Lum/='3') CYCLE
    CALL Aitoff(rad(s.RADeg),rad(s.DEDeg),x,y);
. . . . .
```

Листинг В.13. Изменения в листинге В.8

```

PROGRAM Main
. . . . .
USE LoadSp
. . . . .
CHARACTER(1) :: Sp(2), Lum
. . . . .
CALL OpenTycho2
CALL LoadSpData
. . . . .
  r=7-FLOOR(s.VT/2+0.5); IF (r<1) r=1
  ! Задание цвета звезды, если известен спектральный класс
  IF (FindSpData(s.TYC1,s.TYC2,s.TYC3,Sp(1),Sp(2),Lum)) THEN
  SELECT CASE (Sp(1) )
    CASE ('W','O','B')
      color=RGBTOINTEGER(0,127,255)
    CASE ('A')
      color=RGBTOINTEGER(255,255,255)
    CASE ('F')
      color=RGBTOINTEGER(255,255,127)
    CASE ('G')
      color=RGBTOINTEGER(255,255,0)
    CASE ('K','R')
      color=RGBTOINTEGER(255,127,0)
    CASE ('M','N','S','C','P')
      color=RGBTOINTEGER(255,0,0)
  END SELECT
  ELSE
    color=RGBTOINTEGER(0,255,0)
  END IF
. . . . .

```

Листинг В.14. Вычисление фотометрического расстояния

```

MODULE PhotoDist
! Вычисление фотометрического расстояния

IMPLICIT NONE

REAL(4), PARAMETER :: Mv_V (0:12) = &
(/ -5.7,-4.0,-1.2,0.6,1.9,2.7,3.5,4.4,5.1,5.9,7.4,8.8,12.3/)
! O5 B0 B5 A0 A5 F0 F5 G0 G5 K0 K5 M0 M5
! 0 1 2 3 4 5 6 7 8 9 10 11 12

REAL(4), PARAMETER :: BV_V(0:12) = &
(/-0.33,-0.30,-0.17,-0.02,0.15,0.30,0.44, &
0.58,0.68,0.81,1.15,1.40,1.64/)

REAL(4),PARAMETER :: Mv_III(0:4) = (/0.9, 0.7,-0.2,-0.4,-0.3/)
! G5 K0 K5 M0 M5
REAL(4),PARAMETER :: BV_III(0:4) = (/0.86,1.00,1.50,1.56,1.63/)

REAL(4), PARAMETER :: Mv_Ia(0:11) = &
(/-6.8,-6.9,-7.0,-7.1,-7.4,-8.0,-8.0,-8.0,-7.9,-7.7,-7.5,-7.0/)
! O5 B0 B5 A0 A5 F0 F5 G0 G5 K0 K5 M0

REAL(4), PARAMETER :: BV_Ia(0:11) = &
(/ -0.31,-0.23,-0.08,0.02,0.09,0.17,0.31,
0.75,1.03,1.25,1.60,1.67/)

CONTAINS

LOGICAL FUNCTION PhotoDistance &
(TClass,SClass,LClass,MagVis,B_V,r)
! Возвращает .true. в случае успешного определения

CHARACTER, INTENT(IN) :: TClass ! Температурный класс
CHARACTER, INTENT(IN) :: SClass ! Температурный подкласс
CHARACTER, INTENT(IN) :: LClass ! Класс светимости
REAL(4), INTENT(IN) :: MagVis ! Визуальная звездная величина
REAL(4), INTENT(IN) :: B_V ! Показатель цвета
REAL(4), INTENT(OUT) :: r ! Результат - расстояние в пк

INTEGER(2) :: SubCl, SpCl, Sp
REAL(4) :: x
INTEGER(2) :: k
REAL(4) :: Mtab,BVtab

PhotoDistance=.false.

IF (LClass/='1' .and. LClass/='3' .and. LClass/='5') RETURN

```

```

SELECT CASE(TClass)
  CASE ('W','O')
    SpCl=0
  CASE ('B')
    SpCl=10
  CASE ('A')
    SpCl=20
  CASE ('F')
    SpCl=30
  CASE ('G')
    SpCl=40
  CASE ('K','R')
    SpCl=50
  CASE ('M','N','S','C','P')
    SpCl=60
  CASE DEFAULT
    RETURN
END SELECT

IF (SClass==' ') THEN
  SubCl=5
ELSE
  SubCl=ICHAR(SClass)-ICHAR('0')
END IF

Sp=SpCl+SubCl

SELECT CASE (LClass)
  CASE ('5')
    IF (Sp<5) Sp=5
    IF (Sp>65) Sp=60
    Sp=Sp-5

  CASE ('3')
    IF (Sp<45) Sp=45
    IF (Sp>65) Sp=60
    Sp=Sp-45

  CASE ('1')
    IF (Sp<5) Sp=5
    IF (Sp>60) Sp=55
    Sp=Sp-5

  CASE DEFAULT
    RETURN
END SELECT

x=Sp/5.0
k=FLOOR(x)
x=x-k

```

```

SELECT CASE (LClass)
  CASE ('5')
    IF (k==12) THEN
      Mtab=Mv_V(k); BVtab=BV_V(k)
    ELSE
      Mtab =Mv_V(k)+x*(Mv_V(k+1)-Mv_V(k))
      BVtab=BV_V(k)+x*(BV_V(k+1)-BV_V(k))
    END IF
  CASE ('3')
    IF (k==4) THEN
      Mtab=Mv_III(k); BVtab=BV_III(k)
    ELSE
      Mtab =Mv_III(k)+x*(Mv_III(k+1)-Mv_III(k))
      BVtab=BV_III(k)+x*(BV_III(k+1)-BV_III(k))
    END IF
  CASE ('1')
    IF (k==11) THEN
      Mtab=Mv_Ia(k); BVtab=BV_Ia(k)
    ELSE
      Mtab =Mv_Ia(k)+x*(Mv_Ia(k+1)-Mv_Ia(k))
      BVtab=BV_Ia(k)+x*(BV_Ia(k+1)-BV_Ia(k))
    END IF
  CASE DEFAULT
    RETURN
END SELECT

r=0.2*(MagVis-Mtab)+1-0.2*3.0*(B_V-BVtab)
r=10.0**r
PhotoDistance=.true.
RETURN

END FUNCTION

END MODULE

Листинг В.15. Распределение звезд Tycho-2 Spectral Type по фотометрическим расстояниям

PROGRAM PhotoDistDistrib
! Распред. звезд Tycho-2 Spectral Type по фотом. расстояниям
USE Tycho2Sp
USE PhotoDist
USE DFPort

IMPLICIT NONE

TYPE(TTycho2Sp) star

CHARACTER(*),PARAMETER :: Res = 'photodist2.txt' ! Выходной файл

```

```

INTEGER(4) :: k = 0
INTEGER(4),PARAMETER :: step = Tycho2SpNumOfStars / 80
INTEGER(2),PARAMETER :: N = 15      ! Размер массивов статистики
REAL(4),PARAMETER :: delta = 100.0  ! Шаг по расстоянию

! Статистики распределения звезд по расстоянию
INTEGER(4) :: Dist_V(0:N)=(/(0,i=0,N)/) ! Главная послед.
INTEGER(4) :: Dist_III (0:N) = (/(0,i=0,N)/) ! Гиганты
REAL(4) :: Mj,BVj ! Звездная величина и B-V в системе Джонсона
REAL(4) r ! Расстояние в пк
INTEGER(4) i
INTEGER(4),PARAMETER :: f = 15
INTEGER(4) :: code

CALL OpenTycho2Sp
DO WHILE (ReadTycho2Sp(star))
  k=k+1; IF (MOD(k,step)==0) code=PUTC('*')
  IF (star.NoVT .OR. star.NoBT) CYCLE
  Mj =star.VT-0.090*(star.BT-star.VT)
  BVj=0.850*(star.BT-star.VT)
  IF (PhotoDistance &
      (star.TClass,star.SClass,star.LClass,Mj,BVj,r)) THEN
    i=floor(r/Delta)
    IF (i>N) i=N

    SELECT CASE (star.LClass)
    CASE('3')
      Dist_III(i)=Dist_III(i)+1
    CASE('5')
      Dist_V(i)=Dist_V(i)+1
    END SELECT

  END IF
END DO

CALL CloseTycho2Sp
! Вывод результатов
OPEN(UNIT=f,FILE=Res)
DO i=0,N
  WRITE(f,"(I4,'-',I4,I7,I7)") &
    FLOOR(i*Delta),FLOOR((i+1)*Delta),Dist_V(i),Dist_III(i)
END DO

CLOSE(UNIT=f)
END PROGRAM

```

Листинг В.16. Построение диаграммы Герципрунга–Рессела

```

PROGRAM HRDiagramm ! Построение HR-диаграммы
USE Tycho2Sp
USE PhotoDist
USE DFLIB

IMPLICIT NONE

! Диапазон по AbsMag
REAL(4), PARAMETER :: MagLow =15.0, MagHi ==-10.0
INTEGER, PARAMETER :: Border = 50 ! Отступ диаграммы от края
INTEGER, PARAMETER :: MaxXY = 599

CHARACTER(7), PARAMETER :: sps = 'OBAFGKM';

REAL(8) :: ScaleMg ! Масштаб по оси абс. зв. величин
REAL(8) :: ScaleSP ! Масштаб по оси спектров

! Для инициализации графики
TYPE (windowconfig) wc
TYPE (xycoord) t
INTEGER(1) :: status1
INTEGER(2) :: status2
INTEGER(4) :: status4

CHARACTER(3) buf ! Буфер для формирования текстовых строк
TYPE(TTycho2Sp) star
REAL(4) :: Mj,BVj ! Звездная величина и B-V в системе Джонсона
REAL(4) :: r ! Расстояние в пк
REAL(4) Mag ! Абсолютная звездная величина
INTEGER(4) :: x,y,x1,y1 ! Экранные координаты
INTEGER(4) :: color ! Цвет точки
INTEGER(2) :: i

! Инициализация графического окна
wc.numxpixels = MaxXY
wc.numypixels = MaxXY
wc.numtextcols = -1
wc.numtextrows = -1
wc.numcolors = -1
wc.title = "HR-diagram"C
wc.fontsize = #0008000C ! 10 X 12

status1 = SETWINDOWCONFIG(wc)
IF (.NOT. status1) status1 = SETWINDOWCONFIG(wc)

status2=INITIALIZEFONTS()
status2=SetFont('t'Arial'h10')

! Определение масштабов по осям
ScaleMg=(MaxXY-2*Border)/(MagLow-MagHi)
ScaleSP=(MaxXY-2*Border)/(LineSp('M','9')-LineSp('O','1'))

```

```

! Отображение обрамляющей рамки
CALL XY('O','1',MagHi,x,y)
CALL XY('M','9',MagLow,x1,y1)
status2=Rectangle($GBORDER,x,y,x1,y1);

! Разметка оси спектров
DO i=1,7
  IF (i/=1) THEN
    CALL XY(Sps(i:i),'0',15.0,x,y)
    CALL MoveTo(x,y,t); status2=LineTo(x,y+3)
    CALL MoveTo(x,y+5,t); CALL OutGText(Sps(i:i) // '0')
  END IF
  CALL XY(Sps(i:i),'5',15.0,x,y)
  CALL MoveTo(x,y,t); status2=LineTo(x,y+3)
  CALL MoveTo(x,y+5,t); CALL OutGText(Sps(i:i) // '5')
END DO

! Разметка оси абс. зв. величин
DO Mag=MagHi,MagLow,5.0
  CALL XY('O','1',Mag,x,y)
  CALL MoveTo(x,y,t); status2=LineTo(x-3,y)
  WRITE(buf,'(I3)') FLOOR(Mag)
  CALL MoveTo(x-15,y,t); CALL OutGText(buf)
END DO

CALL OpenTycho2SP
DO WHILE (ReadTycho2Sp(star))
  IF (star.NoVT .or. star.NoBT) CYCLE
  Mj =star.VT-0.090*(star.BT-star.VT);
  BVj =0.850*(star.BT-star.VT);
  IF (.not. PhotoDistance(star.TClass,star.SClass,star.LClass, &
    Mj,BVj,r)) CYCLE
  Mag=Mj-5.0*log10(r)+5.0 ! Вычисл. абс. звезд. величины
  CALL XY(star.TClass,star.SClass,Mag,x,y);
  SELECT CASE (star.LClass)
  CASE ('1')
    color=RGBTOINTEGER(0,0,255)
  CASE ('2')
    color=RGBTOINTEGER(0,255,255)
  CASE ('3')
    color=RGBTOINTEGER(255,0,0)
  CASE ('4')
    color=RGBTOINTEGER(255,255,0)
  CASE ('5')
    color=RGBTOINTEGER(0,255,0)
  END SELECT
  status4=SetColorRGB(color)
  status2=SetPixel(x,y)
END DO
CALL CloseTycho2Sp

status4=SaveImage("HR.bmp"C,0,0,MaxXY,MaxXY)

```

```

CONTAINS ! Внутренние процедуры

INTEGER FUNCTION LineSp(Sp,SubSp)
CHARACTER :: Sp,SubSp
INTEGER(2) :: SpCl

SELECT CASE(Sp)
  CASE('W','O')
    SpCl=0
  CASE('B')
    SpCl=10
  CASE('A')
    SpCl=20
  CASE('F')
    SpCl=30
  CASE('G')
    SpCl=40
  CASE('K','R')
    SpCl=50
  CASE('M','N','S','C','P')
    SpCl=60
  CASE DEFAULT
    SpCl=-9999
END SELECT
IF (SubSp==' ') SubSp='5'
LineSp=SpCl+ICHAR(SubSp)-ICHAR('0')
END FUNCTION

! Процедура пересчета показателя цвета и абс. зв. величины
! в экранные координаты

SUBROUTINE XY(Sp,SubSp,Mag,x,y)
CHARACTER, INTENT(IN) :: Sp,SubSp
REAL(4), INTENT(IN) :: Mag
INTEGER(4), INTENT(OUT) :: x,y
  x=Border+FLOOR(LineSp(Sp,SubSp)*ScaleSP+0.5)
  y=Border+FLOOR((Mag-MagHi)*ScaleMg+0.5)
END SUBROUTINE

END PROGRAM

```

Листинг В.17. Сравнение тригонометрических расстояний звезд каталога *Hipparcos* с фотометрическими расстояниями, полученными на основе каталога *Tycho-2 Spectral Types*

```

PROGRAM COMPAREDISTANCES
USE HipMain
USE Tycho2
USE PhotoDist
USE LoadSp
USE DFPort

INTEGER(4), PARAMETER :: NHip = 117955; ! Звезд с параллаксами
REAL(4), PARAMETER :: RelErr = 0.5; ! Относ. ош. параллакса
REAL(4), PARAMETER :: Rmax = 500.0; ! Максимальное расстояние

TYPE THip
  INTEGER(4) :: Hip ! Номер звезд в Hipparcos
  REAL(4) :: Plx ! Параллакс звезды в mas
  REAL(4) :: SPlx ! Ошибка параллакса в mas
END TYPE

TYPE (THip) HipCat(NHip)

INTEGER(4) :: step = Tycho2NumOfStars/80

INTEGER(4) :: k
TYPE (TTycho2) :: s
INTEGER(4) :: h! Номер элемента массива в Hipparcos
REAL(4) :: Mj,BVj ! Звездная величина и B-V в системе Джонсона
CHARACTER(1) :: TClass,SClass,LClass ! Спектральный класс
REAL(4) :: r_px,r_ph ! Триг. и фотометрическое расстояния в ПК
INTEGER(4) :: f = 22 ! Файл результатов

PRINT *, 'Load Hipparcos'
CALL LoadHipparcos
PRINT *, 'Sort Hipparcos'
CALL SortHipparcos
CALL OpenTycho2
PRINT *, 'Load Spectral Data'
CALL LoadSpData

OPEN (UNIT=f, FILE='r.txt')

```

```

DO WHILE (ReadTycho2(s))

  k=k+1; IF (MOD(k,step)==0) code=PUTC('*')

  ! Поиск звезды в Hipparcos
  h=FindHip(s.Hip);
  IF (h==0) CYCLE ! Если звезда не найдена, пропустить
  IF (HipCat(h).Plx<=0.0) CYCLE ! "Плохой" параллакс
  ! Плохая точность параллакса
  IF (HipCat(h).sPlx/HipCat(h).Plx>RelErr) CYCLE
  r_px=1000.0/HipCat(h).Plx;
  IF (r_px>Rmax) CYCLE
  ! Проверка по звездной величине
  IF (s.NoVT .or. s.NoBT) CYCLE
  ! Поиск спектральных данных
  IF (.not. FindSpData(s.TYC1,s.TYC2,s.TYC3, &
    TClass,SClass,LClass)) CYCLE

  ! Вычисление фотометрического расстояния
  Mj =s.VT-0.090*(s.BT-s.VT); BVj=0.850*(s.BT-s.VT);
  IF (.not. PhotoDistance(TClass,SClass,LClass,Mj,BVj,r_ph)) &
    CYCLE

  IF (r_ph>Rmax) CYCLE
  WRITE(f, '(I6,2F8.1)') HipCat(h).Hip,r_px,r_ph
END DO

CLOSE (UNIT=f)

CONTAINS

SUBROUTINE LoadHipparcos;
! Загрузка данных каталога Hipparcos в массив HipCat
INTEGER(4) :: i
TYPE (THipparcos) s
CALL OpenHipparcosMain
i=0
DO WHILE (ReadHipparcosMain(s))
  IF (s.NoPlx) CYCLE
  i=i+1
  HipCat(i).Hip=s.Hip
  HipCat(i).Plx=s.Plx
  HipCat(i).SPlx=s.sigma_Plx
END DO
CALL CloseHipparcosMain
END SUBROUTINE

```

```

SUBROUTINE SortHipparcos
! Пузырьковая сортировка массива HipCat

LOGICAL Sorted ! флаг - массив отсортирован
INTEGER(4) :: l, r, i
TYPE(THip) t

l=1; r=NHip
DO WHILE(.not. Sorted)
  Sorted=.True. ! Установка флага
  DO i=l,r-1 ! Проход слева направо
    IF (HipCat(i).Hip>HipCat(i+1).Hip) THEN
      t=HipCat(i); HipCat(i)=HipCat(i+1); HipCat(i+1)=t
      Sorted=.False. ! Сброс флага
    END IF
  END DO
  r=r-1

  DO i=r-1,1,-1 ! Проход справа налево
    IF (HipCat(i).Hip>HipCat(i+1).Hip) THEN
      t=HipCat(i); HipCat(i)=HipCat(i+1); HipCat(i+1)=t
      Sorted=.False. ! Сброс флага
    END IF
  END DO
  l=l+1

END DO

END SUBROUTINE

INTEGER(4) FUNCTION FindHip(Hip0)
! Бинарный поиск элемента массива, в котором Hip=Hip0
INTEGER(4) :: Hip0
INTEGER(4) :: l, r, n

FindHip=0
l=1; r=NHip
DO WHILE (L<R)
  n=(l+r)/2
  IF (HipCat(n).Hip==Hip0) THEN
    FindHip=n;
    EXIT
  END IF
  IF (HipCat(n).Hip<Hip0) THEN
    l=n+1
  ELSE
    r=n
  END IF
END DO
END FUNCTION

END PROGRAM

```

Литература

- Perryman M.A.C. et al. The Hipparcos and Tycho Catalogues, Vol.1-17, ESA, 1997.
- Høg E. Tycho-2 CD-ROM, 2003.
- Høg E. et al. Guide to the Tycho-2 Catalogue. Copenhagen Univ. Obs. 2000.
- Høg E. et al. The Tycho-2 Catalogue of the 2.5 Million Brightest Stars // A.&A., 2000, 355, L27.
- Høg E. et al., Construction and Verification of the Tycho-2 Catalogue // A.&A., 2000, 357, 367.
- <http://vizier.u-starsbg.fr/cgi-bin/VizieR?-source=Tycho-2>
- <http://aladin.u-strasbg.fr/>
- <http://archive.eso.org/skycat/servers/ASTROM>
- <http://www.astro.ku.dk/~erik/Tycho-2>
- Цветков А.С. Руководство по практической работе с каталогом Hipparcos, СПб.: СПбГУ, 2005.
- <http://www.freepascal.org>
- Попов А.В., Цветков А.С. Звездный состав каталога Tycho-2 // Вестн. С.-Петерб. ун-та, 2005.
- Perryman M.A.C. et al. The Hipparcos and Tycho Catalogues. Vol.1, part 1.3, 2.2. ESA, 1997.
- Perryman M.A.C. et al. The Hipparcos and Tycho Catalogues. Vol.1, part 1.4. ESA, 1997.
- Perryman M.A.C. et al. The Hipparcos and Tycho Catalogues. Vol.1, p.91. ESA, 1997.
- Киселев А.А. Теоретические основания фотографической астрометрии, 1989. М.: Наука. С. 62.
- RedShift 4. Менеджер проекта: О.Белайчук; консультанты: С.Кудрявцев, К.Лоутон, Дж.Милтон, В.Сурдин, В.Шевченко; программисты: А.Жуков, В.Афанасьев, А.Алексеев, Н.Биткин, А.Добровольский, С.Фатахов, А.Куликов, И.Логинов, К.Маргорин, О.Маргорин, А.Титов, В.Улитетин; оформление: Т.Киряева, HTML: Н.Калитцева, Р.Кумушкулов; звук: С.Усанов.
- Wright C.O., Egan M.P., Kraemer K.E., Price S.D., The Tycho-2 Spectral Type Catalog // Astron. J. 2003. 125, 359.
- <http://cdsweb.u-strasbg.fr/cgi-bin/Cat?III/231>
- Houk N. et al. Michigan Catalog of Two-dimensional Spectral Types for HD Stars. Vol. 1-5. Ann. Arbor: Univ. Michigan Dept. Astron. 1978, 1982, 1975, 1988, 1999.

21. Jaschek C., Conde H., de Sierra A.C. Catalog of Stellar Spectra Classified in the Morgan-Keenan System. La Plata: La Plata Obs. 1964.
22. Kennedy P.M. MK Classification Catalog Extention. Weston Creek: Mt. Stromlo & Siding Spring Obs. 1983.
23. Fricke W. et. al. Fifth Fundamental Catalog Part I, II. Heidelberg: Veröffentlich. Astron. Rechen-Inst. 1988, 1991.
24. Bastian U., Röser S. Catalog of Positions and Proper Motions. Heidelberg: Astron. Rechen-Inst. South – 1993; North – 1998.
25. Lang K.R., Astrophysical Data: Planets and Stars. New York: Springer, 1992.
26. Fabricius et al. Каталог IV/25 в базе CDS. 2002.
27. Schmidt-Kaler Th. Landolt-Bornstein: Numerical data and Functional Relationships in Science and Technology. Vol. 2b / Eds. K. Schaifers, H.H. Voigt, Berlin: Springer. 1982.
28. Binney J., Merrifield M. Galactic Astronomy. Princeton: Princeton University Press, 1998.
29. Becker, F. Bonner Durchmusterung, Noerdlicher Teil, Deklinations-Zonen -1 bis +89 grade Sternverzeichnis, dritte, berichtigte Auflage. Bonn: Ferd. Duemmlers Verlag, 1951.
30. Cannon A.J., Pickering E.C. HD Catalogue. Ann. Harvard Obs., 91-99. 1918-1924.
31. <http://www.astro.spbu.ru>
32. Turon C. et al. The Hipparcos Input Catalogue. // Bull. Inform. CDS, Vol. 43. P.5.
33. Вирт Н. Алгоритмы и структуры данных. СПб.: Невский диалект, 2001.

Учебное издание

Александр Станиславович Цветков

РУКОВОДСТВО ПО РАБОТЕ
С КАТАЛОГОМ ТУСНО-2

Учебно-методическое пособие

Зав. редакцией *Г.И.Чердниченко*
Оформление обложки *Е.А.Соловьевой*
Оригинал-макет *А.С.Цветкова*

Подписано в печать с оригинала-макета 01.12.2005.
Ф-т 60x84/16. Усл. печ. л. 7,67. Уч.-изд. л. 7,5. Тираж 70 экз.
Заказ №

РОПИ С.-Петербургского государственного университета.
199034, С.-Петербург, Университетская наб., 7/9.

Типография Издательства СПбГУ.
199061, С.-Петербург, Средний пр., 41.