

Федеральное агентство по образованию  
Уральский государственный технический университет – УПИ  
имени первого Президента России Б. Н. Ельцина

Электронный аналог печатного издания

# **ВЫПОЛНЕНИЕ КУРСОВОЙ РАБОТЫ ПО ДИСЦИПЛИНЕ «ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ ВЫСОКОГО УРОВНЯ»**

Методические указания для студентов специальности 230150  
«Программное обеспечение вычислительной техники  
и автоматизированных систем»

Екатеринбург  
УГТУ – УПИ  
2008

УДК 004.2:004.9

Составитель С. И. Тимошенко, канд. техн. наук

Научный редактор В. И. Суханов, д-р техн. наук

**ВЫПОЛНЕНИЕ КУРСОВОЙ РАБОТЫ ПО ДИСЦИПЛИНЕ «ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ ВЫСОКОГО УРОВНЯ»** : методические указания для студентов специальности 230150 «Программное обеспечение вычислительной техники и автоматизированных систем» / сост. С. И. Тимошенко. — Екатеринбург: УГТУ – УПИ, 2008. — 21 с.

В методических указаниях описана структура и содержание разделов пояснительной записки к курсовой работе. Приведены варианты индивидуальных заданий к курсовой работе.

Библиогр.: 8 назв.

Подготовлены кафедрой  
«Программные средства и системы»  
факультета ускоренного обучения

# Содержание

3

---

## Содержание

Введение	4
1 Структура и содержание пояснительной записки	5
1.1 Титульный лист	5
1.2 Задание на курсовую работу	5
1.3 Содержание	6
1.4 Введение	6
1.5 Постановка задачи	6
1.6 Анализ поставленной задачи	7
1.7 Описание результатов разработки	7
1.8 Руководство пользователя	7
1.9 Заключение	7
1.10 Список использованных источников	7
1.11 Приложение	8
2 Варианты задач	9
2.1 Задача о канадских авиалиниях	9
2.2 Задача о расстановке ферзей	9
2.3 Задача о шахматном коне	9
2.4 Задача о лабиринте	9
2.5 Задача о рюкзаке	10
2.6 Задача о коммивояжере	10
2.7 Задача о парламенте	10
2.8 Задача об автозаправке	10
2.9 Задача о паркете	11
2.10 Задача о черепашке	11
2.11 Задача о почтовых отделениях	11
2.12 Задача о диалоге компьютеров	12
2.13 Задача о системах счисления	13
2.14 Задача о копилке	13
2.15 Задача об игре умножения	13
2.16 Задача «Анти-QuickSort»	13
2.17 Задача о строках Фибоначчи	13
2.18 Задача о пути спелеолога	14
2.19 Задача о гирлянде	14
2.20 Задача о формировании поезда	14
2.21 Задача «Network»	15
2.22 Задача о метро	15
2.23 Задача о веревочном телеграфе	16
2.24 Задача «Secret Pipes»	16
2.25 Задача о кубиках	17
2.26 Задача о кодировании	17
2.27 Задача о мобильных телефонах	18
2.28 Задача о проверке веб-страниц	18
2.29 Задача об обмене валют (Currency Exchange)	19
2.30 Задача об обмене товаров (Exchange Rates)	19
Список использованных источников	20

## Введение

Курсовая работа по дисциплине «Программирование на языке высокого уровня» имеет целью систематизацию теоретических и практических знаний студентов в области программирования, развитие навыков самостоятельной работы, освоение технологий тестирования программных средств, обучение методам выбора и обоснования программных решений, изучение современных стандартов.

В состав курсовой работы входят пояснительная записка и комплект программ, решающих поставленную задачу.

Правила оформления пояснительной записки содержатся в методических указаниях [1] и основаны на стандартах ГОСТ 2.105-95, ГОСТ 7.1-2003 и ГОСТ 7.32-2001.

Уточним, что страницы пояснительной записки должны иметь формат А4. Правое поле должно быть 10 мм, верхнее, левое и нижнее — 20 мм. Текст должен быть набран шрифтом с кеглем не менее 12 пунктов, межстрочным интервалом 150 % от кегля.

Основной текст пояснительной записки должен быть набран гарнитурой с засечками (например, Times, Georgia или Minion Pro). Заголовки разделов, подразделов и пунктов — рубленой гарнитурой жирных начертаний (например, Arial, Verdana, Tahoma или Myriad Pro). Исходный текст программ — моноширинной гарнитурой (например, Courier).

В первом разделе методических указаний рассмотрены требования к структуре и содержанию пояснительной записки.

Во втором разделе приведен список из тридцати вариантов задач. Вариант задачи для курсовой работы студент вытягивает по билету.

В конце методических указаний приведен список использованных источников, которые необходимы в том числе и для выполнения курсовой работы. В работе [1] описаны общие правила оформления курсовых. В работах [2, 3] описаны концепции экстремального программирования. В работах [4, 5, 6] приведены полные описания задач. Электронные ресурсы [7, 8] содержат информацию для любознательных.

# 1 Структура и содержание пояснительной записки

5

1

**Структура  
и содержание  
пояснительной  
записки**

Структура пояснительной записки курсовой работы по дисциплине «Программирование на языках высокого уровня» следующая:

- титульный лист;
- задание на курсовую работу;
- содержание;
- введение;
- постановка задачи;
- анализ поставленной задачи;
- описание результатов разработки;
- руководство пользователя;
- заключение;
- список использованных источников;
- приложение.

## 1.1 Титульный лист

Титульный лист является первым листом пояснительной записки, но номер листа на нем не проставляется. Шаблон титульного листа можно взять у руководителя. Обычно на титульном листе сверху указывается федеральное агентство, ниже полное наименование учебного заведения, затем название кафедры, место для оценки работы. Ниже темы работы необходимо указать инициалы и фамилии руководителя и студента. Перед защитой на титульном листе должны быть поставлены подписи всех вышеуказанных лиц с простановкой даты подписи.

## 1.2 Задание на курсовую работу

Задание на курсовую работу составляется по установленной форме, которую можно взять у руководителя. Группа студентов разбивается на пары. Каждой паре выдается свое задание. Распределение ролей в паре может меняться для отработки навыков парного программирования (это один из принципов экстремального программирования Кента Бека [2, 3]). Однако формально каждый из участников отвечает за разные части проекта: один за разработку основного модуля решения задачи, второй — за разработку модуля тестирования с использованием JUnit.

В задании на курсовую работу это будет отражено во второй части названия. Например, для первого студента из пары оно будет звучать так: «Разработка программы для решения задачи о коммивояжере. Основной модуль». Для второго студента из пары оно запишется несколько иначе: «Разработка программы для решения задачи о коммивояжере. Модуль тестирования». Задание на курсовую работу не нумеруется.

### **1.3 Содержание**

Содержание должно включать введение, наименование всех разделов, подразделов, пунктов, заключение, список использованных источников, наименование приложений с указанием номеров страниц, с которых начинаются эти элементы. Если для написания пояснительной записки используется текстовый процессор, формируйте содержание автоматически. Возможно, придется это делать многократно по мере внесения в текст исправлений и дополнений. Заголовок раздела «Содержание» не нумеруется и выравнивается по центру.

### **1.4 Введение**

Введение должно содержать краткое изложение решаемой задачи, а также пояснение назначения и цели создания программы. Во введении должны быть указаны методы и инструментальные средства, используемые для решения задач курсовой работы. В конце введения необходимо указать структуру пояснительной записки с кратким описанием содержания каждого раздела. Заголовок раздела «Введение» не нумеруется и выравнивается по центру.

### **1.5 Постановка задачи**

В разделе необходимо подробно описать решаемую задачу. Это не повтор задания на курсовую работу, а развернутое описание (с приведением вспомогательных рисунков и таблиц) самой задачи и формулировка требований к программе, которая будет ее решать. Типичной ошибкой при написании этого раздела является замена подробного описания ссылкой на источник (или несколько источников), где такое описание есть.

## 1.6 Анализ поставленной задачи

В этом разделе нужно провести анализ требований к разрабатываемой программе, привести возможные алгоритмы решения поставленной задачи и выбрать наилучший из них. Необходимо обосновать структуру данных и архитектуру разрабатываемой программы, описать схему работы системы по ГОСТ 19.701-90.

## 1.7 Описание результатов разработки

Раздел должен содержать описание двух модулей: основного модуля, который решает задачу, и модуля тестирования (с использованием JUnit). Описание модулей должно сопровождаться диаграммами классов по правилам языка UML, а также фрагментами кода и рисунками (скриншотами) результатов работы и тестирования.

## 1.8 Руководство пользователя

В разделе необходимо указать, какие ресурсы компьютера необходимы для работы вашей программы, как установить программу на компьютер, чтобы она правильно функционировала, как удалить программу и вспомогательные файлы. Кроме того, нужно дать инструкцию по работе с программой, сопроводив ее скриншотами по этапам работы. Необходимо описать также тестовые варианты входных и выходных данных.

## 1.9 Заключение

Раздел должен отражать основные результаты проделанной работы, содержать оценку полноты решения поставленной задачи, рекомендации по использованию полученных результатов и улучшению программы. Заголовок раздела «Заключение» не нумеруется и выравнивается по центру.

## 1.10 Список использованных источников

В списке указываются источники, использованные в процессе работы. Сведения об источниках должны соответствовать требованиям ГОСТ 7.1-2003 и ГОСТ 7.82-2001. Заголовок раздела «Список использованных источников» не нумеруется и выравнивается по центру.

## **1.11 Приложение**

В приложении необходимо привести в полном объеме исходный текст программы. В некоторых случаях можно поместить вспомогательный справочный материал.

Заголовок раздела «Приложение» нумеруется буквами и выравнивается по центру [1]. Ниже заголовка в круглых скобках указывается слово «обязательное», а в следующей строке — название приложения с большой буквы (например, «Исходный текст программы»).



## 2 Варианты задач

9

2

Варианты  
задач

### 2.1 Задача о канадских авиалиниях

Вы победили в соревновании, организованном канадскими авиалиниями. Приз — бесплатное путешествие по Канаде. Путешествие начинается с самого западного города, в который летают самолеты, проходит с запада на восток, пока не достигнет самого восточного города, в который летают самолеты. Затем путешествие продолжается обратно с востока на запад, пока не достигнет начального города. Ни один из городов нельзя посещать более одного раза за исключением начального города, который надо посетить ровно дважды (в начале и в конце путешествия). Вам также нельзя пользоваться авиалиниями других компаний или другими способами передвижения. Задача состоит в следующем: дан список городов и список прямых рейсов между парами городов; найти маршрут, включающий максимальное количество городов и удовлетворяющий выше-названным условиям.

Пояснения к задаче можно посмотреть в [4, с. 121-124].

### 2.2 Задача о расстановке ферзей

На шахматной доске  $N \times N$  требуется расставить  $N$  ферзей таким образом, чтобы ни один ферзь не атаковал другого.

Пояснения к задаче можно посмотреть в [4, с. 81-86].

### 2.3 Задача о шахматном коне

Существуют способы обойти шахматным конем доску, побывав на каждом поле по одному разу. Составить программу подсчета числа способов обхода.

Пояснения к задаче можно посмотреть в [4, с. 86-89].

### 2.4 Задача о лабиринте

Дано клеточное поле, часть клеток занята препятствиями. Необходимо попасть из некоторой заданной клетки в другую заданную клетку путем последовательного перемещения по клеткам.

Пояснения к задаче можно посмотреть в [4, с. 89-92].

## 2.5 Задача о рюкзаке

В рюкзак загружаются предметы  $N$  различных типов (количество предметов каждого типа не ограничено). Максимальный вес рюкзака  $W$ . Каждый предмет типа  $i$  имеет вес  $w_i$  и стоимость  $v_i$  ( $i=1, 2, \dots, N$ ). Требуется определить максимальную стоимость груза, вес которого равен  $W$ .

Пояснения к задаче можно посмотреть в [4, с. 92-93, 102-105].

## 2.6 Задача о коммивояжере

Имеются  $N$  городов, расстояния между которыми заданы; коммивояжеру необходимо выйти из какого-то города, посетить остальные  $N-1$  городов точно по одному разу и вернуться в исходный город. При этом маршрут коммивояжера должен быть минимальной длины (стоимости).

Пояснения к задаче можно посмотреть в [4, с. 93-96].

## 2.7 Задача о парламенте

На острове Новой Демократии каждый из жителей организовал партию, которую сам и возглавил. Любой из жителей острова может состоять не только в своей партии, но и в других партиях. Ко всеобщему удивлению, даже в самой малочисленной партии оказалось не менее двух человек. К сожалению, финансовые трудности не позволили создать парламент, куда вошли бы, как предполагалось по Конституции острова, президенты всех партий. Посовещавшись, островитяне решили, что будет достаточно, если в парламенте будет хотя бы один член каждой партии. Помогите островитянам организовать такой, как можно более малочисленный парламент, в котором будут представлены все партии.

Пояснения к задаче можно посмотреть в [4, с. 113-116].

## 2.8 Задача об автозаправке

Вдоль кольцевой дороги расположено  $M$  городов, в каждом из которых есть автозаправочная станция. Известна стоимость  $Z[i]$  заправки в городе с номером  $i$  и стоимость  $C[i]$  проезда по дороге, соединяющей  $i$ -й и  $(i+1)$ -й города,  $C[M]$  — стоимость проезда между первым и  $M$ -м городами. Для жителей каждого города определить город, в который им необходимо съездить, чтобы заправиться самым дешевым образом, и

направление — «по часовой стрелке» или «против часовой стрелки» (города пронумерованы по часовой стрелке).

Пояснения к задаче можно посмотреть в [4, с. 117-118].

11

2

Варианты  
задач

## 2.9 Задача о паркете

Комнату размером  $N \times M$  единиц требуется покрыть одинаковыми плитками паркета размером  $2 \times 1$  единиц без пропусков и наложений ( $M < 20$ ,  $N < 8$ ,  $M$ ,  $N$  — целые). Пол можно покрыть паркетом различными способами. Требуется определить количество всех возможных способов укладки паркета для конкретных значений  $M < 20$ ,  $N < 8$ . Результат задачи — таблица, содержащая 20 строк и 8 столбцов. Элементом таблицы является число, являющееся решением задачи для соответствующих  $N$  и  $M$ . На месте ненайденных результатов должен стоять символ «\*».

Пояснения к задаче можно посмотреть в [4, с. 119-121].

## 2.10 Задача о черепашке

Черепашка находится в городе, все кварталы которого имеют прямоугольную форму, и ей необходимо попасть с крайнего северо-западного перекрестка на крайний юго-восточный. На некоторых улицах проводится ремонт, и по ним запрещено движение (например, между перекрестками 3 и 7, 5 и 6, 10 и 11), длина, а значит и стоимость проезда по остальным улицам задается. Кроме того, для каждого перекрестка определена стоимость поворота. Так, если Черепашка пришла на 7-й перекресток и поворачивает к 11-му, то она платит штраф, а если идет в направлении 8-го, то платить ей не приходится. Найти для Черепашки маршрут минимальной стоимости.

Пояснения к задаче можно посмотреть в [4, с. 126-127].

## 2.11 Задача о почтовых отделениях

Вдоль прямой дороги расположены деревни. Дорога представляется целочисленной осью, а расположение каждой деревни задается одним целым числом — координатой на этой оси. Никакие две деревни не имеют одинаковых координат. Расстояние между двумя деревнями вычисляется как модуль разности из координат. В некоторых, не обязательно во всех, деревнях будут построены почтовые отде-

ления. Деревня и расположенное в ней почтовое отделение имеют одинаковые координаты. Почтовые отделения необходимо расположить в деревнях таким образом, чтобы общая сумма расстояний от каждой деревни до ближайшего к ней почтового отделения была минимальной. Напишите программу, которая по заданным координатам деревень и количеству почтовых отделений находит такое расположение почтовых отделений по деревням, при котором общая сумма расстояний от каждой деревни до ее ближайшего почтового отделения будет минимальной.

Пояснения к задаче можно посмотреть в [4, с. 134-137].

## **2.12 Задача о диалоге компьютеров**

Три компьютера соединены сетью. Один из них — сервер, два других — клиенты. На сервере есть несколько файлов. Полные имена файлов, состоящие из двух частей (имя и расширение), различны. Оба клиента знают полные имена всех файлов, находящихся на сервере. Сервер выбирает один из своих файлов и посылает его имя одному из клиентов, а расширение — второму. Затем клиенты начинают общаться друг с другом, пытаясь определить, какой файл был выбран сервером (они хотят узнать полное имя файла). Однако клиенты вынуждены общаться очень ограниченным способом. Они по очереди посылают сообщения друг другу, но могут сказать только, что не знают полного имени файла.

Если клиент не знает полного имени выбранного файла, он может послать другому клиенту сообщение, говорящее: «Я не знаю полного имени файла». Клиенты чередуются, посылая только это сообщение туда и обратно. Так продолжается до тех пор, пока один из клиентов не узнает полное имя файла или они не решат закончить диалог. Клиент, получивший первую часть полного имени файла, всегда ждет, что второй клиент пошлет первое сообщение.

Пусть вы знаете все полные имена файлов, находящихся на сервере, и слушаете разговор клиентов. Основываясь на этой беседе, вы должны определить набор файлов, которые могли быть выбраны сервером. Файлы в этом наборе называются файлами-кандидатами.

Пояснения к задаче можно посмотреть в [5, с. 56-57, 199-202].

## 2.13 Задача о системах счисления

13

Дано целое неотрицательное число в  $I$ -ричной системе счисления. Вывести это число в  $J$ -ричной системе счисления.

2

Варианты  
задач

Пояснения к задаче можно посмотреть в [5, с. 40, 155-158].

## 2.14 Задача о копилке

Заданы вес  $E$  пустой копилки и вес  $F$  копилки с монетами. В копилке могут находиться монеты  $N$  видов; известны ценность  $P_i$  каждого вида монет и вес  $W_i$  одной монеты. Найти минимальную и максимальную суммы денег, которые могут находиться в копилке.

Пояснения к задаче можно посмотреть в [5, с. 33, 126-128].

## 2.15 Задача об игре умножения

Слава и Оля играют в игру умножения — умножают целое число  $P$  на одно из чисел от 2 до 9. Слава всегда начинает с  $P = 1$ , делает умножение, затем число умножает Оля, затем Слава и т. д. Перед началом игры им задают случайное число  $N$ , и победителем считается тот, кто первым получит  $P > N$ . Определить, кто выиграет при заданном  $N$ , если оба играют наилучшим образом.

Пояснения к задаче можно посмотреть в [5, с. 47, 173-176].

## 2.16 Задача «Анти-QuickSort»

Для сортировки последовательности чисел широко используется быстрая сортировка — QuickSort. Хотя QuickSort является самой быстрой сортировкой в среднем, существуют тесты, на которых она работает очень долго. Требуется написать программу, генерирующую тест, на котором быстрая сортировка сделает наибольшее число сравнений.

Пояснения к задаче можно посмотреть в [5, с. 62-63, 217-219].

## 2.17 Задача о строках Фибоначчи

Строку Фибоначчи  $F(K)$  для натуральных чисел  $K$  определим так:  $F(1) = 'A'$ ,  $F(2) = 'B'$ ,  $F(K) = F(K - 1) + F(K - 2)$  при  $K > 2$ , где «+» означает конкатенацию строк. Требуется найти ко-

личество вхождений строки  $S$ , состоящей из символов  $A$  и  $B$ , в строку Фибоначчи  $F(N)$ .

Пояснения к задаче можно посмотреть в [5, с. 63, 219-221].

## 2.18 Задача о пути спелеолога

Пещера представлена кубом, разбитым на  $N$  частей по каждому измерению (то есть на  $N^3$  кубических клеток). Каждая клетка может быть или пустой, или полностью заполненной камнем. Исходя из положения спелеолога в пещере, требуется найти, какое минимальное количество перемещений по клеткам ему требуется, чтобы выбраться на поверхность. Переходить из клетки в клетку можно, только если они обе свободны и имеют общую грань.

Пояснения к задаче можно посмотреть в [5, с. 65, 225-226].

## 2.19 Задача о гирлянде

Гирлянда состоит из  $N$  лампочек на общем проводе. Один ее конец закреплен на заданной высоте  $A$  мм ( $H_1 = A$ ). Благодаря силе тяжести гирлянда прогибается: высота каждой неконцевой лампы на 1 мм меньше, чем средняя высота ближайших соседей ( $H_i = (H_{i-1} + H_{i+1})/2 - 1$ ,  $1 < i < N$ ). Требуется найти минимальную высоту второго конца  $B$  ( $B = H_N$ ) при условии, что ни одна из лампочек не должна лежать на земле ( $H_i > 0$ ,  $1 \leq i \leq N$ ).

Пояснения к задаче можно посмотреть в [5, с. 71-72, 240-243].

## 2.20 Задача о формировании поезда

Компания, занимающаяся железнодорожными перевозками, получила заказ сформировать поезд, состоящий из определенного числа вагонов. Проблема заключается в том, что у компании есть вагоны, выпущенные в разное время, так что каждый из вагонов может иметь один из двух видов сцепления на каждом конце. У компании также есть один локомотив. Сцепления и для локомотива, и для вагонов обозначены буквой  $A$  или  $B$ . Повернуть вагон или локомотив противоположной стороной невозможно.

Дана информация о вагонах и локомотиве. Требуется найти число способов сформировать разные поезда заданной длины из имеющихся видов вагонов. Дополнительным требованием является то, что тип сцеплений на каждом конце состава должен соответствовать типу сцепления локомотива. Поезда считаются различными, если при их сравнении от одного конца к другому выявляется хотя бы одно отличие.

Пояснения к задаче можно посмотреть в [5, с. 88-89, 281-289].

## 2.21 Задача «Network»

Андрей работает системным администратором и планирует создание новой сети в своей компании. Всего будет  $N$  хабов, они будут соединены друг с другом с помощью кабелей. Поскольку каждый сотрудник компании должен иметь доступ ко всей сети, каждый хаб должен быть достижим от любого другого хаба — возможно, через несколько промежуточных хабов. Поскольку имеются кабели различных типов и короткие кабели дешевле, требуется сделать такой план сети (соединения хабов), чтобы максимальная длина одного кабеля была как можно меньшей. Есть еще одна проблема — не каждую пару хабов можно непосредственно соединять по причине проблем совместимости и геометрических ограничений здания. Андрей снабдит вас всей необходимой информацией о возможных соединениях хабов. Вы должны помочь Андрею найти способ соединения хабов, который удовлетворит всем указанным выше условиям.

Пояснения к задаче можно посмотреть в [6, с. 61-62, 73-75, 51-53].

## 2.22 Задача о метро

В некотором городе есть метро, состоящее из  $N$  ( $1 \leq N \leq 1000$ ) станций и  $M$  ( $0 \leq M \leq 500000$ ) линий, соединяющих их. Каждая линия обеспечивает проезд между какими-то двумя станциями в обе стороны. Между любой парой станций проведено не более одной линии. Сеть метро построена таким образом, чтобы с каждой станции можно было проехать на каждую (возможно, через промежуточные станции). Назовем это свойство связностью метро. В связи с изобретением принципиально

нового вида транспорта метро стало убыточным, и его работу решили прекратить. На заседании мэрии города было постановлено закрывать каждый год по одной станции, но так, чтобы связность метро каждый раз сохранялась. При закрытии какой-либо станции линии, ведущие от этой станции к другим, естественно, тоже перестают функционировать.

Пояснения к задаче можно посмотреть в [6, с. 59-61, 71-73, 51-53].

## 2.23 Задача о веревочном телеграфе

Тимур и его друзья, приехав летом на свои старые дачи, решили устроить на время своего отдыха игру. Они организовали команду, чтобы тайно помогать жителям дачного городка в их повседневных делах. Дачный поселок довольно большой, и дома, в которых живут друзья Тимура, расположены далеко друг от друга. Как быстро передавать друг другу сообщения? Как собирать ребят на совет? Тимур решил проложить веревочный телеграф, который связал бы все домики, в которых живут ребята из его команды. Всего домиков  $N$ . По карте ребята вычислили координаты каждого домика  $(X_i, Y_i)$  в целых числах и выписали их на бумаге. За единицу измерения координат они взяли один метр. Однако возник вопрос: какие домики нужно соединять веревочным телеграфом, чтобы связь была между всеми домиками (возможно, через другие домики), а общая длина всех веревок была как можно меньше? Требуется написать программу, которая по координатам домиков определяла бы, какова минимальная общая длина всех веревок, соединяющих все домики между собой (возможно, через другие домики).

Пояснения к задаче можно посмотреть в [6, с. 47-53, 63-64].

## 2.24 Задача «Secret Pipes»

Фермер Джон хочет как можно дешевле организовать свою систему распределения воды, но он не хочет, чтобы его конкурент фермер Плуту мог предсказать маршруты, которые он выбирает. Фермер Джон знает, что такая задача обычно требует самого дешевого способа прокладки труб. Поэтому он решил использовать второй по стоимости способ. Дан список всех двунаправ-



ленных труб, которые могут соединять множество из  $W$  ( $3 \leq W \leq 2\,000$ ) станций с водой (каждая из которых может быть встроена в колодец). Ваша задача — найти второй из самых дешевых способов соединить насосные станции, используя не более чем  $P$  ( $P \leq 20\,000$ ) труб с заданной стоимостью каждой трубы. Не должно быть трубы, соединяющей станцию саму с собой. Не должно быть двух труб, соединяющих дважды одну и ту же пару станций. Гарантируется, что есть только один самый дешевый способ распределить воду, и что существует, как минимум, два способа распределить воду. Все стоимости — положительные числа, помещающиеся в 16-битное целое. Водная станция идентифицируется своим номером — целым числом в диапазоне  $1..W$ .

Пояснения к задаче можно посмотреть в [6, с. 55-59, 67-70].

## 2.25 Задача о кубиках

Родители подарили Пете набор детских кубиков. Поскольку Петя скоро пойдет в школу, они купили ему кубики с буквами. На каждой из шести граней каждого кубика написана буква. Теперь Петя хочет похвастаться перед старшей сестрой, что научился читать. Для этого он хочет сложить из кубиков ее имя. Но это оказалось довольно сложно сделать — ведь разные буквы могут находиться на одном и том же кубике, и тогда Петя не сможет использовать обе буквы в слове. Правда, одна и та же буква может встречаться на разных кубиках. Помогите Пете!

Пояснения к задаче можно посмотреть в [6, с. 18-21, 35-37].

## 2.26 Задача о кодировании

Необходимо произвести оптимальное кодирование входного текста, представляющего собой набор символов из множества  $M = \{', ; , , , ' , '0', '9', 'A', 'Z', 'a', 'z', 'A', 'Я', 'a', 'я'\}$  в двоичное представление. Это значит, что каждый символ входного предложения должен быть закодирован уникальным набором нулей и единиц. Символы, не принадлежащие множеству  $M$ , кодировать и помещать в выходной файл не надо. Выходные данные, представляющие собой закодированный текст, также должны однозначно декодироваться. В данном случае оптимальным кодом называется код, содержащий наименьшее суммарное число единиц и нулей.

Пояснения к задаче можно посмотреть в [6, с. 120-124, 171-173].

## 2.27 Задача о мобильных телефонах

Предположим, что базовые станции для мобильных телефонов 4-го поколения, расположенные в районе Тампере, функционируют следующим образом: район поделен на квадраты. Квадраты образуют матрицу  $S \times S$ , в которой строки и столбцы нумеруются от 0 до  $S - 1$ . Каждый квадрат содержит базовую станцию. Количество активных мобильных телефонов внутри квадрата может изменяться, потому что телефоны перемещаются из квадрата в квадрат или потому, что телефоны включаются и выключаются. Время от времени каждая базовая станция сообщает изменения в своем количестве активных телефонов на главную базовую станцию, единственную для всех строк и столбцов матрицы. Напишите программу, которая получает эти отчеты и отвечает на запросы о текущем общем количестве активных мобильных телефонов в любой области прямоугольной формы.

Пояснения к задаче можно посмотреть в [6, с. 137-141, 179-181].

## 2.28 Задача о проверке веб-страниц

Многим из тех, кому приходилось работать в Интернете, случалось сталкиваться с неправильными ссылками, то есть ссылками на несуществующие документы. Ваша задача — реализовать упрощенную проверку страницы на корректность ссылок. Входной файл содержит название одного или нескольких документов и их содержимое. Содержимое документов имеет следующий вид:

```
<HTML>  
Текст  
<END>
```

В тесте могут присутствовать ссылки на другие документы на данном сервере. Они имеют следующий вид:  
<A HREF=«имя файла»>.

Пояснения к задаче можно посмотреть в [6, с. 225-230, 279-281].

## 2.29 Задача об обмене валют (Currency Exchange)

19

2

Варианты  
задач

В городе работает несколько пунктов обмена валют. Каждый пункт специализируется на обмене двух конкретных валют и выполняет обменные операции только над ними. Может быть несколько пунктов, специализирующихся на обменах одной и той же пары валют. Каждый пункт имеет свой собственный обменный курс. Обменный курс валюты А к валюте В есть количество валюты В, которое выдается за единицу валюты А. Кроме того, каждый пункт обмена назначает «комиссионные» — сумму, которую вы должны заплатить за операцию обмена. Комиссия всегда собирается в исходной валюте. Николай имеет некоторое количество денег в валюте S и хочет узнать, может ли он некоторой последовательностью операций обмена увеличить свой капитал. Конечно, он хочет иметь в конце операций деньги в той же исходной валюте S. Помогите ему ответить на этот трудный вопрос. Николай всегда должен иметь неотрицательную сумму денег во время выполнения операций обмена.

Пояснения к задаче можно посмотреть в [6, с. 211-215, 272-274].

## 2.30 Задача об обмене товаров (Exchange Rates)

Использование денег для оплаты за товары и услуги обычно упрощает жизнь, но иногда люди предпочитают обмениваться непосредственно товарами. В таком случае торговцы устанавливают коэффициенты обменов между товарами. Коэффициент обмена между двумя товарами А и В выражается двумя положительными целыми числами  $m$  и  $n$ . В этом случае говорят, что  $m$  предметов товара А равноценны  $n$  предметам товара В. Например, 2 печи могут быть равноценны 3 холодильникам. Ваша задача — написать программу, которая по заданному списку обменных коэффициентов вычислит обменный коэффициент между двумя заданными товарами.

Пояснения к задаче можно посмотреть в [6, с. 215-220, 274-277].

## **Список использованных источников**

- 1 Оформление курсовых и дипломных проектов [Текст]: методические указания для студентов технических специальностей / сост. В. Н. Кичигин, И. Е. Мясников, С. И. Тимошенко. — Екатеринбург: ГОУ ВПО «УГТУ-УПИ», 2005. — 80 с.
- 2 Бек К. Экстремальное программирование [Текст] / К. Бек. — СПб. : Питер, 2002. — 224 с.
- 3 Бек К. Экстремальное программирование [Текст]: разработка через тестирование / К. Бек. — СПб. : Питер, 2003. — 224 с.
- 4 Окулов С. М. Программирование в алгоритмах [Текст] / С. М. Окулов. — М. : БИНОМ. Лаборатория знаний, 2004. — 341 с.
- 5 Меньшиков Ф. В. Олимпиадные задачи по программированию [Текст] / Ф. В. Меньшиков. — СПб. : Питер, 2006. — 315 с.
- 6 Долинский М. С. Решение сложных и олимпиадных задач по программированию [Текст] : учебное пособие / М. С. Долинский. — СПб. : Питер, 2006. — 366 с.
- 7 Алгоритмы, методы, исходники [Электронный ресурс]. — 29.03.2007. — Режим доступа: <http://algolist.manual.ru/>.
- 8 Густокашин М. С. Разбор олимпиадных задач по информатике [Электронный ресурс] / М. С. Густокашин. — 29.03.2007. — Режим доступа: <http://g6prog.narod.ru/>.

*Учебное издание*

**ВЫПОЛНЕНИЕ КУРСОВОЙ РАБОТЫ  
ПО ДИСЦИПЛИНЕ «ПРОГРАММИРОВАНИЕ  
НА ЯЗЫКЕ ВЫСОКОГО УРОВНЯ»**

Составитель     Сергей Иванович Тимошенко

Редактор Е. В. Рябая

ИД № 062063 от 12.11.2001 г.

Подписано в печать 18.01.07		Формат 60x84 1/16
Бумага 80 г/м	Цифровая печать	Усл. печ. л. 1,16
Уч.-изд. л. 1.0	Тираж 100	Заказ 154

Редакционно-издательский отдел УГТУ – УПИ  
620002, г. Екатеринбург, ул. Мира, 19  
rio@fat.ustu.ru

Отпечатано в отделении полиграфии ИВТОБ  
620002, Екатеринбург, ул. Мира, 19, ауд. И-120  
Тел. (343) 375-41-43  
opivtob@mail.ustu.ru