

Министерство образования и науки Российской Федерации  
Федеральное государственное учреждение высшего образования  
Московский государственный университет технологий и управления  
(первый казачий университет)  
Университетский колледж информационных технологий

Специальность 09.02.03 Программирование в компьютерных системах

ПМ1 Разработка программных модулей программного обеспечения компьютерных сетей

МДК1.2 Прикладное программирование Сборник заданий для самостоятельной работы (1 часть)

## Содержание

<b>1</b>	<b>Аннотация</b>	<b>3</b>
<b>2</b>	<b>Самостоятельная работа №1. Программная документация (11 часов)</b>	<b>5</b>
2.1	Общие сведения . . . . .	5
2.2	Теоретическое введение . . . . .	5
2.2.1	Техническое задание . . . . .	6
2.2.2	Программа и методика испытаний . . . . .	10
2.2.3	Пояснительная записка технического проекта . . . . .	17
2.2.4	Программа автоматического тестирования модуля . . . . .	20
2.2.5	Программа . . . . .	20
2.2.6	Протокол испытаний . . . . .	20
2.3	Варианты заданий . . . . .	20
<b>3</b>	<b>Самостоятельная работа №2. Указатели и списки (7 часов)</b>	<b>23</b>
<b>4</b>	<b>Самостоятельная работа №3. ООП (6 часов)</b>	<b>26</b>
<b>5</b>	<b>Самостоятельная работа №4. Конспект (3 часа)</b>	<b>28</b>
<b>6</b>	<b>Самостоятельная работа №5. Видеоурок «простейшие компоненты» (3 часа)</b>	<b>28</b>
<b>7</b>	<b>Самостоятельная работа №6. Доклады и видеоуроки: простейшие компоненты, работающие с графической информацией (3 часа)</b>	<b>29</b>
<b>8</b>	<b>Самостоятельная работа №7. Доклад и видеоурок: компоненты для работы с большими наборами данных (3 часа)</b>	<b>30</b>
<b>9</b>	<b>Самостоятельная работа №8. Паттерны проектирования (3 часа)</b>	<b>30</b>
<b>10</b>	<b>Самостоятельная работа №9. Эссе (1 час)</b>	<b>32</b>
<b>11</b>	<b>Список литературы</b>	<b>32</b>

# 1 Аннотация

Сборник заданий содержит задания для самостоятельной работы по МДК «Прикладное программирование» для первой половины курса (40 часов самостоятельных работ).

МДК «Прикладное программирование» входит в профессиональный модуль «ПМ1 Разработка программных модулей программного обеспечения компьютерных сетей» и направлен на выполнение следующих требований к студенту<sup>1</sup>:

- иметь практический опыт:
  - разработки алгоритма поставленной задачи и реализации его средствами автоматизированного проектирования;
  - разработки кода программного продукта на основе готовой спецификации на уровне модуля;
  - использования инструментальных средств на этапе отладки программного продукта;
  - проведения тестирования программного продукта по определенному сценарию;
- уметь:
  - осуществлять разработку кода программного модуля на современных языках программирования;
  - создавать программу по разработанному алгоритму как отдельный модуль;
  - выполнять отладку и тестирование программы на уровне модуля;
  - оформлять документацию на программные средства;
  - использовать инструментальные средства для автоматизации оформления документации;
- знать
  - основные этапы разработки программного обеспечения;
  - основные принципы технологии структурного и объектно-ориентированного программирования;
  - основные принципы отладки и тестирования программных продуктов;
  - методы и средства разработки технической документации.

Результатом освоения профессионального модуля должно быть формирование следующих компетенций:

- ПК1.1 выполнять разработку спецификаций отдельных компонент;
- ПК1.2 осуществлять разработку кода программного продукта на основе готовых спецификаций на уровне модуля;
- ПК1.3 выполнять отладку программных модулей с использованием специализированных программных средств;
- ПК1.4 выполнять тестирование программных модулей;
- ПК1.5 осуществлять оптимизацию программного кода модуля;
- ПК1.6 разрабатывать компоненты проектной и технической документации с использованием графических языков спецификаций.

Кроме того, изучение профессионального модуля должно способствовать формированию следующих общих компетенций:

- ОК1 понимать сущность и социальную значимость своей будущей профессии, проявлять к ней устойчивый интерес;
- ОК2 организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество;
- ОК3 принимать решения в стандартных и нестандартных ситуациях и нести за них ответственность;
- ОК4 осуществлять поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития;
- ОК5 использование информационно-коммуникационных технологий в профессиональной деятельности;

<sup>1</sup>Федеральный государственный стандарт среднего специального образования по специальности 09.02.03 Программирование в компьютерных системах, утвержденный приказом Министерства образования и науки Российской Федерации от 28 июля 2014 г. №804

- ОК6 работать в коллективе и в команде, эффективно общаться с коллегами, руководством, потребителями;
- ОК7 брать на себя ответственность за работу членов команды (подчиненных), за результат выполнения заданий;
- ОК8 самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации;
- ОК9 ориентироваться в условиях частой смены технологий в профессиональной деятельности;

Новая редакция сборника отличается разработкой дополнительного числа вариантов с учетом численности групп, унификацией заданий по теме «списки», внедрением нового методического приема – разработка видеоуроков.

## 2 Самостоятельная работа №1. Программная документация (11 часов)

### Цели работы

- получение опыта разработки алгоритма (в том числе средства автоматизированного проектирования);
- разработка кода на основе готовой спецификации;
- проведение тестирования по определенному сценарию;
- создание программы по разработанному алгоритму;
- оформление документации на программные средства;
- закрепление знаний основных этапов разработки программного обеспечения;
- закрепление знаний методов и средств разработки технической документации;
- проработка профессиональных компетенций ПК1.1-1.2, 1.4, 1.6;
- проработка общих компетенций ОК2-5, 6.

### 2.1 Общие сведения

Целью самостоятельной работы является освоение создания части программной документации для программных изделий.

Особенностью самостоятельной работы является то, что в ходе её выполнения происходит передача уже сделанной части работы другому студенту: таким образом каждый вариант делают до шести различных студентов. Каждая часть выполняется одну неделю, в случае задержки работы на срок больше недели (кроме случая болезни) все части в дальнейшем выполняет один студент, но оценка понижается за всю работу до тройки.

Самостоятельная работа состоит из следующих частей:

1. Составление, согласование и утверждение технического задания
2. Составление, согласование и утверждение программы и методики испытаний
3. Составление, согласование и утверждение технического проекта
4. Написание программы, предназначенной для автоматического тестирования модуля
5. Написание собственно программы (с учетом отладки и тестирования)
6. Проведение и протоколирование испытаний

Второе и третье задание выдается одновременно двум разным студентам (для одной программы). Четвертое и пятое задание также выдается одновременно двум разным студентам, но студент выполняющий задание номер пять перед сдачей должен получить программу задания №4.

Сдача происходит следующим путем:

1. Вы высылаете свою работу на e-mail: mgkit@rsc.pp.ru (проверка осуществляется, в основном, по понедельникам).
2. После проверки и исправления всех замечаний вы распечатываете (или пишете от руки) документ
3. Вы сдаете напечатанную или написанную работу преподавателю.

### 2.2 Теоретическое введение

Промышленная разработка программных изделий почти всегда сопровождается процессом документирования, то есть оформлением в виде документов хода разработки и результатов этапов разработки программы. В нашей стране (в основном в случае участия в государственном заказе) используются стандарты, разработанные еще во времена СССР. Их модификацию мы и будем использовать в учебных целях.

В этой сфере используются две серии государственных стандартов: 19 и 34. Первая из них описывает документирование собственно программных продуктов, а вторая – автоматизированных систем (представляющих собой совокупность технической базы, программного обеспечения, обслуживающего персонала и т. д.). Разумеется, мы будем использовать ГОСТы 19-ой серии.

ГОСТы регламентируют не только содержание документации, но и шрифты, правила оформления и так далее, которые мы не будем соблюдать в полной мере (чтобы не погрязнуть в не столь важных деталях).

Мы будем использовать следующую модель первых этапов жизненного цикла (адаптированная версия ГОСТ 19.102-77):

1. Разработка технического задания (соответствует работе «анализ требований к программным средствам» в ISO 12207)
2. Технический проект (соответствует работам «проектирование программной архитектуры» и «техническое проектирование программных средств»)
3. Рабочий проект
  - (а) Разработка программы (в ISO 12207 соответствует работе «программирование и тестирование программных средств»)
  - (б) Испытания программы («сборка программных средств» и «квалификационное испытание программных средств»)

ГОСТ дает достаточно широкую свободу в перечне создаваемых документов, и мы среди них выберем следующие (некоторые другие будут изучены в рамках 3 профессионального модуля):

- Техническое задание
- Пояснительная записка (стадия технического проекта)
- Текст программы (этап разработки программы)
- Программа и методика испытаний (этап испытания программы)

В ряде случаев (например, для текста программы) мы будем использовать локальные стандарты (в рамках курса «Прикладное программирование»).

*Далее, мы пройдем все этапы создания программного изделия на примере простой программы – решение квадратного уравнения. В самостоятельной работе вам будет предложено пройти те же этапы при создании программы примерно той же сложности. В практических работах вы будете создавать весь комплект документов для достаточно сложных программных изделий.*

### 2.2.1 Техническое задание

Техническое задание – это документ, призванный чётко охарактеризовать тот объем работы, что должен выполнить исполнитель. В сфере разработки программного обеспечения техническое задание чётко, полно, непротиворечиво должно описывать будущий программный продукт и требования к ходу его разработки (например, используемые языки программирования, перечень создаваемых документов).

**Общие сведения об оформлении** У каждого документа есть титульный лист. Мы его будем оформлять следующим образом (здесь мы не следуем ГОСТ):

Министерство образования и науки Российской Федерации  
ФГБОУ ВО МГУТУ им. К. Г. Разумовского (ПКУ)  
Университетский колледж информационных технологий

## **РЕШЕНИЕ КВАДРАТНОГО УРАВНЕНИЯ** **Техническое задание**

**ПП-00**

**Листов 20**

Исполнитель  
Иванов И. И.  
30.02.2015

Другие документы имеют аналогичный по оформлению титульный лист (меняется только его название).  
Разделы и подразделы нумеруются в форматах 1 и 1.1

**Содержание технического задания** Техническое задание состоит из разделов:

1. **Введение.** Указывается:

- Название;
- область применения (для простейших программ название и область применения отличаются только формулировкой);
- объект, где используется программное изделие (имеется ввиду предприятие, где будет использоваться разработанное программное изделие).

1. Введение  
 1.1 Наименование программы  
 Решение квадратного уравнения  
 1.2 Область применения  
 Деятельность по решению квадратных уравнений  
 1.3 Объект, в котором используют программу  
 Программа «решение квадратных уравнений» сможет быть использована в любых объ-  
 ектах

2. **Основание для разработки:**

- документ, на основании которого ведется разработка;
- организация, утвердившая этот документ, и дата утверждения;
- наименование и (или) условное обозначение темы для разработки; (мы будем использовать обозначение *ПП-номер варианта*)

2. Основание для разработки  
 2.1 Документ, на основании которого ведется разработка  
 Сборник самостоятельных работ по МДК01.02 Прикладное программирование  
 2.2 Наименование и (или) условное обозначение темы для разработки  
 ПП-0

3. **Назначение разработки:**

- функциональное назначение (здесь нужно ответить на вопрос «зачем нужна программа?»);
- эксплуатационное назначение (здесь нужно ответить на вопрос «для кого нужна программа?»)

3. Назначение разработки  
 3.1 Функциональное назначение  
 Решение квадратных уравнений  
 3.2 Эксплуатационное назначение  
 Программа предназначена для использования неограниченным кругом лиц

4. **Требования к программе или программному изделию**

- **требования к функциональным характеристикам** (состав функций, организация входных и выходных данных, временные характеристики и др.)

В этом разделе как можно более подробно нужно описать все возможные действия, которые пользователь может совершить с использованием программы. Функции можно описать в иерархическом виде, описывая сначала какую-то функцию целиком, а потом в подпунктах описать те подфункции, на которые функция распадается.

Помните о том, что все требования должны быть однозначно понимаемыми, выполнимыми в совокупности, проверяемыми (должна быть возможность ответить на вопрос: выполнено требование или нет при предъявлении программы). Ну и, естественно, они должны соответствовать ожиданиям заказчика.

Входные данные нужно описывать как можно более подробно, чтобы при проверке соответствия программы техническому заданию нельзя было ввести входные данные в достаточно произвольном виде.



Выходные данные лучше описать наиболее обобщенно (по способу их представления) и как можно точнее (по набору выходных данных).

Следует иметь в виду, что вы имеете право вводить дополнительные разделы с целью лучшей структуризации и более четкого описания программного изделия.

#### 4. Требования к программе или программному изделию

##### 4.1 Требования к функциональным характеристикам

###### 4.1.1 Требования к составу выполняемых функций

Программа должна обеспечивать выполнение следующих функций:

- решение квадратного уравнения

###### 4.1.2 Требования к организации входных данных

Входные данные читаются из стандартного потока ввода. Вещественные числа должны соответствовать формату литералов вещественных чисел языка Паскаль, каждое из которых не превышает по модулю  $10^{18}$ .

###### 4.1.3 Требования к организации выходных данных

Выходные данные отправляются в стандартный поток вывода и представляют собой одну из следующих строк: «Некорректные входные данные», «Данное уравнение не имеет решений», «Данное уравнение имеет один корень  $x=####$ », «Данное уравнение имеет два корня:  $x1=####$ ,  $x2=####$ », где вместо  $####$  подставляются числа.

###### 4.1.4 Требования к временным характеристикам

Время между окончанием ввода и окончанием вывода результата не должно превышать одной секунды.

- **Требования к надежности** (надежного функционирования: обеспечения устойчивого функционирования, контроль входной и выходной информации, время восстановления после отказа и т.п.)

Требования к надежности нужно разрабатывать исходя из того, что для разработчика лучше, если требований к надежности будет как можно меньше, а для заказчика – как можно больше.

##### 4.2 Требования к надежности

###### 4.2.1 Требования к контролю входной информации

4.2.1.1 Контролировать соответствие формата входной информации на синтаксическое соответствие литералам вещественных чисел языка Паскаль не требуется.

4.2.1.2 Контролировать отсутствие переполнения при чтении входной информации не требуется.

4.2.1.3 Требуется проверять старший коэффициент на равенство нулю с выводом соответствующего сообщения.

4.2.2 Другие требования к надежности не предъявляются

- **Условия эксплуатации** (включая квалификацию персонала)

Условия эксплуатации четко должны описывать условия, при которых программа будет корректно работать.

##### 4.3 Условия эксплуатации

4.3.1 Минимальное количество персонала – один человек, владеющий русским языком, обладающий практическими навыками по использованию программ, работающих в режиме командной строки, знающий тему «Квадратные уравнения» курса математики.

4.3.2 Другие специальные требования к условиям эксплуатации не предъявляются

- **Требования к составу и параметрам технических средств**

##### 4.4 Требования к составу и параметрам технических средств

В состав технических средств должен входить компьютер, включающий:

- процессор Pentium или совместимый с ним;
- клавиатуру;
- видеокарту;
- монитор;
- жесткий или SSD-диск.

- **Требования к информационной и программной совместимости** (требования к информационным структурам на входе и выходе и методам решения, исходным кодам, языкам программирования и программным средствам, используемым программой)

В данном разделе, в частности, могут описываться форматы файлов, используемые операционные системы, библиотеки.

#### 4.5 Требования к информационной и программной совместимости

4.5.1 Требования к информационным структурам на входе и выходе не предъявляются

4.5.2 Требования к методам решения.

Решать квадратное уравнение требуется с использованием обычных формул, использующих дискриминант.

4.5.3 Требования к исходным кодам изложены в документе: А. И. Глускер «Сборник задач по курсу “Основы алгоритмизации и программирования” [Электронный ресурс] – 2011 (раздел 3.1)

4.5.4 Программа должна быть написана на языке Паскаль и компилироваться транслятором Free Pascal 2.6

4.5.5 Программа должна работать под управлением операционной системы Windows XP.

4.5.6 Требования к защите информации и программ не предъявляются

5. **Требования к программной документации** (состав и возможно требования: здесь мы не будем следовать ГОСТу полностью)

#### 5 Требования к программной документации

##### 5.1 Состав программной документации

Состав программной документации должен включать:

- техническое задание;
- пояснительную записку;
- текст программы;
- текст программы, осуществляющей автоматическое тестирование программы «решение квадратного уравнения»;
- программу и методику испытаний.

##### 5.2 Специальные требования к пояснительной записке

Пояснительная записка должна содержать блок-схему (блок-схемы) алгоритма(-ов), используемых в программе.

6. **Стадии и этапы разработки** (стадии, этапы, содержание работ: сроки и исполнители)

#### 7 Стадии и этапы разработки

##### 7.1 Стадии разработки

Разработка осуществляется в три стадии:

- техническое задание;
- технический проект;
- рабочий проект.

##### 7.2 Этапы разработки

На стадии техническое задание осуществляется разработка, согласование и утверждение технического задания в срок до 26 февраля 2016 года Глускером А. И.

На стадии технический проект осуществляется разработка, согласование и утверждение пояснительной записки в срок до 5 марта 2016 года Глускером А. И.

На стадии рабочий проект осуществляется разработка текста программы, осуществляющей автоматическое тестирование программы «решение квадратного уравнения», разработка, согласование и утверждение программы и методики испытаний, текста программы в срок до 7 марта 2016 года Глускером А. И., после чего осуществляются испытания по результатам которой возможно будет проводиться корректировка программной документации в срок до 12 марта 2016 года.

7. **Порядок контроля и приемки**

#### 8 Порядок контроля и приемки

Приемосдаточные испытания должны проводиться в соответствии с программой и методикой испытаний, разработанной, согласованной и утвержденной не позднее 7 марта 2016 года.

### 2.2.2 Программа и методика испытаний

Программа и методика испытаний – это документо подробно, полно и однозначно описывающий способ проверки соответствия разработанного (в будущем) программного изделия техническому заданию. Программа

и методика испытаний содержит описание того способа этой проверки, на котором сошлись обе стороны: и заказчик, и исполнитель.

Первая часть программы и методики испытаний содержит выписки из технического задания, содержащие общие сведения о программном изделии и требования к нему. Далее, приводится порядок проведения испытаний и ссылки на методики, которые при этом используются. В конце документа приводятся методики проверки для каждого этапа испытаний.

Следует учитывать, что в процессе проведения испытаний ведется протокол, где отражаются результаты проведенных испытаний.

## 1. Объект испытаний

### 1.1 Наименование

Решение квадратного уравнения

### 1.2 Область применения

Деятельность по решению квадратных уравнений

### 1.3 Обозначение программы

ПП-0

## 2. Цель испытаний

Проверка соответствия программного продукта требованиям технического задания.

## 3. Требования к программе

### 3.1 Требования к функциональным характеристикам

#### 3.1.1 Требования к составу выполняемых функций

Программа должна обеспечивать выполнение следующих функций (п. 4.1.1 технического задания):

- функция решения квадратного уравнения

#### 3.1.2 Требования к организации выходных данных

Выходные данные отправляются в стандартный поток вывода и представляют собой одну из следующих строк: «Некорректные входные данные», «Данное уравнение не имеет решений», «Данное уравнение имеет один корень  $x=####$ », «Данное уравнение имеет два корня:  $x1=####$ ,  $x2=####$ », где вместо  $####$  подставляются числа (п. 4.1.3 технического задания)

#### 3.1.3 Требования к временным характеристикам

Время между окончанием ввода и окончанием вывода результата не должно превышать одной секунды (п. 4.1.4 технического задания)

### 3.2 Требования к надежности

Требуется проверять старший коэффициент на равенство нулю с выводом соответствующего сообщения (п. 4.2.1.1 технического задания)

### 3.3 Требования к информационной и программной совместимости

Программа должна работать под управлением операционной системы Windows XP (п. 4.5.5 технического задания)

## 4. Требования к программной документации

### 4.1 Состав программной документации

Состав программной документации должен включать (п. 5.1 технического задания):

- техническое задание;
- пояснительную записку;
- текст программы;
- текст программы, осуществляющей автоматическое тестирование программы «решение квадратного уравнения»;
- программу и методику испытаний.

### 4.2 Специальные требования к пояснительной записке

Пояснительная записка должна содержать блок-схему (блок-схемы) алгоритма(-ов), используемых в программе (п. 5.2 технического задания)

### 4.3 Специальные требования к тексту программы

4.3.1 Требования к исходным кодам изложены в документе: А. И. Глускер «Сборник задач по курсу “Основы алгоритмизации и программирования” [Электронный ресурс] – 2011 /раздел 3.1/ (п. 4.5.3 технического задания)

4.3.2 Программа должна быть написана на языке Паскаль и компилироваться транслятором Free Pascal 2.6 (п. 4.5.4 технического задания)

В разделах «требования к программе» и «требования к программной документации» можно вместо переписывания требований просто сослаться на них (по номерам пунктов в техническом задании). В списке указываются только требования, которые подлежат проверке.

В разделе «Средства и порядок испытаний» указываются технические и программные средства, используемые во время проведения испытаний, а также порядок проведения испытаний.

## 5. Средства и порядок испытаний

### 5.1 Технические средства, используемые при проведении испытаний

В состав технических средств входит IBM-совместимый компьютер, включающий:

- процессор Pentium или совместимый с ним;
- клавиатуру;
- видеокарту;
- монитор;
- жесткий или SSD-диск.

### 5.2 Программные средства, используемые при проведении испытаний

В состав программных средств входит:

- лицензионная копия операционной системы Windows XP (любой версии), локализованная для работы в Российской Федерации;
- FreePascal 2.6 или более поздняя версия (вариант, предназначенный для работы в среде Windows).

### 5.3 Порядок проведения испытаний

5.3.1 Подготовка к проведению испытаний заключается в обеспечении наличия компьютера, описанного в п. 5.1, и программных средств, указанных в п. 5.2, установленных на этом компьютере

5.3.2 Ход проведения испытаний документируется в протоколе, где указывается перечень проводимых испытаний, результат каждого испытания и возможно замечания.

5.3.3 Состав испытаний:

5.3.3.1 Проверка состава программной документации в соответствии с методом, описанном в п. 6.2

5.3.3.2 Проверка требований к программе

Проверка обеспечения требований к программе (п. 3) в соответствии с методом, описанным в п. 6.1

5.3.3.3 Проверка требований к программной документации

5.3.3.3.1 Проверка пояснительной записки (п. 4.2) в соответствии с методом, описанным в п. 6.3

5.3.3.3.2 Проверка текстов программ (п. 4.3.1) в соответствии с методом, описанным в п. 6.4

5.3.3.3.3 Проверка текстов программ (п. 4.3.2) в соответствии с методом, описанным в п. 6.5

Состав испытаний может быть более или менее подробным в зависимости от испытываемого программного продукта и характера взаимосвязей между требованиями.

В разделе «методы испытаний» подробно описывается способ проведения их. В частности, указываются тестовые примеры, ожидаемые ответы.

## 6. Методы испытаний

### 6.1 Метод проверки требований к программе

Проверка осуществляется путем запуска программы и сравнения результатов ее запуска с ожидаемыми, при этом осуществляется измерения интервала времени между окончанием ввода и окончанием вывода результатов.

При каждом запуске программы осуществляется ввод трех чисел ( $a$ ;  $b$  и  $c$ ) и визуально контролируется совпадение результатов с ожидаемыми (допускается отклонение чисел в результирующих строках не более, чем на 0,00001, числа могут быть представлены в любом формате, допустимом для литералов языка Паскаль).

Список тестовых примеров:

№	Входные данные			Выходные данные
1	0	1	1	Некорректные входные данные
2	1	0	0	Данное уравнение имеет один корень $x=0$
3	1	2	1	Данное уравнение имеет один корень $x=-1$
4	1	2	0	Данное уравнение имеет два корня $x_1=0$ , $x_2=-2$
5	2	4	0	Данное уравнение имеет два корня $x_1=0$ , $x_2=-2$
6	1	3	2	Данное уравнение имеет два корня $x_1=-1$ , $x_2=-2$
7	-1	-3	-2	Данное уравнение имеет два корня $x_1=-1$ , $x_2=-2$
8	1	2	4	Данное уравнение не имеет решений
9	-1	-2	-4	Данное уравнение не имеет решений

В тестовых примерах №№4–7 допускается другой порядок корней в выходных данных.

- В случае, если все результаты тестовых примеров №№2–9 совпадают с контрольными за исключением, возможно, конкретных формулировок, то в протокол вносится запись: «Требования к составу выполняемых функций» – соответствует; в противном случае: «Требования к составу выполняемых функций» – не соответствует.
- В случае, если все результаты тестовых примеров имеют формат, указанный в п. 3.1.2, то в протокол вносится запись: «Требования к организации выходных данных» – соответствует; в противном случае: «Требования к организации выходных данных» – не соответствует.
- В случае, если результат тестового примера №1 совпадает с контрольным за исключением, возможно, конкретной формулировки, то в протокол вносится запись: «Требования к надежности» – соответствует; в противном случае: «Требования к надежности» – не соответствует.
- В случае, если длина временного интервала от окончания ввода входных данных до окончания вывода результатов, не превышает одной секунды для каждого тестового примера, то в протокол вносится запись: «Требования к временным характеристикам» – соответствует; в противном случае «Требования к временным характеристикам» – не соответствует.
- В случае, если при выполнении хотя бы одного тестового примера осуществляется запуск программы и ожидается ввод входных данных, то в протокол вносится запись: «Требования к информационной и программной совместимости в части п. 4.5.5 технического задания» – соответствует; в противном случае «Требования к информационной и программной совместимости в части п. 4.5.5 технического задания» – не соответствует.

### 6.2 Метод проверки требований к составу программной документации

Проверка состава программной документации осуществляется визуально путем сравнения набора предъявленных документов (в форме распечатки или в рукописной форме) списку, приведенному в п. 4.1. При этом исходные тексты программ должны быть предоставлены так же и в электронной форме.

В случае если набор предъявленных документов соответствует списку, а исходные тексты предоставлены также в электронной форме, то в протокол заносится запись: «Состав программной документации» – соответствует; в противном случае: «Состав программной документации» – не соответствует.

### 6.3 Метод проверки требований к пояснительной записке

Проверка состоит из следующих этапов:

- проверка наличия блок-схемы (блок-схем) в пояснительной записке;
- проверка соблюдения требований ГОСТ 19.701-90 для каждой блок-схемы;
- проверка соблюдения локальных стандартов для блок-схем;
- проверка соответствия каждой блок-схемы алгоритму, закодированному в программе.

Проверка соблюдения требований ГОСТ 19.701-90 состоит из следующих работ:

- проверка использования только тех символов, которые указаны как применимые к схемам программ в п. 5 ГОСТ 19.701-90;
- проверка соответствия символов их назначению (экспертная оценка лица, проводящего испытание);
- проверка правильности выполнения соединения линий (п. 4.2.3 ГОСТ 19.701-90);
- проверка того, что линии потока управления, выходящие из символа «решение» подписана (п. 4.3.1.2 ГОСТ 19.701-90);

Проверка соблюдения локальных стандартов для блок-схем состоит из следующих работ:

- проверка того, что все символы (кроме терминаторов, соединителей, линий и комментариев) имеют одинаковые размеры;
- проверка того, что терминаторы имеют ту же ширину, что и другие символы;
- проверка того, что отношение ширины к высоте составляет 2 к 1 для каждого символа, кроме терминаторов, комментариев и линий
- проверка того, что отношение ширины к высоте составляет 4 к 1 для терминаторов;
- проверка того, что высота соединителей совпадает с высотой терминаторов;
- проверка того, что линии потока управления входят в символ слева или сверху, а выходят снизу или справа;
- проверка того, что подписи к линиям не находятся на самих линиях.

Проверка соответствия каждой блок-схемы алгоритму, закодированному в программе, осуществляется путем экспертной оценки лицом, осуществляющим проведение испытаний.

В случае, если все вышеприведенные проверки прошли успешно, в протокол заносится запись: «Специальные требования к пояснительной записке» – соответствует; в противном случае «Специальные требования к пояснительной записке» – не соответствует.

#### 6.4 Метод проверки требований к исходным кодам

Изложенный ниже метод применяется ко всем файлам, содержащим исходный текст, и входящим в состав программной документации по отдельности. Для каждого файла вносится в протокол запись: «Требования к исходным кодам для файла #####» – соответствует/не соответствует (где вместо ##### указывается название файла).

Проверка состоит из следующих этапов:

- Наличие комментария в начале файла, содержащего автора работы, номера задания и варианта, краткой формулировки задания (или его части)
- Наличие комментариев к неочевидным действиям (проверяется методом экспертной оценки лицом, осуществляющим испытания)
- Для каждой подпрограммы наличие комментария, содержащего полное описание ее работы, описание всех аргументов и результатов. Достаточность этого комментария для возможности использовать подпрограмму в других программах (без изучения собственно текста подпрограммы).
- Для каждой глобальной переменной указание ее назначения.
- Для всех переменных, кроме переменных цикла, использование «говорящих» названий
- Для всех подпрограмм использование говорящих названий.
- Использование одного оператора на одной строке программы.
- Количество пробелов перед строкой программы должно соответствовать уровню вложенности (по два пробела на уровень вложенности).
- Слова begin и end, соответствующие друг другу, располагаются строго с одной и той же позиции по вертикали.
- Количество строк в подпрограмме и в самой программе (между begin и end) – не более 25 строк.
- Использование модулей для трех и более сходных по назначению подпрограмм.
- Отсутствие в подпрограммах использования глобальных переменных (напрямую).
- Разделение подпрограмм на предназначенные для вычислений (в них не должно быть ввода-вывода) и на предназначенные для ввода-вывода (в них вычисления должны быть только такие, что нужны для ввода-вывода).
- Отсутствие операторов goto, break, continue; процедур halt и exit.
- Проверка того, что вместо явно указанных значений чисел, в тексте программы используются константы.

В случае, если все перечисленные этапы пройдены, то в протокол о соответствии файла требованиям, в противном случае – о несоответствии.

#### 6.5 Метод проверки требований к исходным кодам в части компиляции во Free Pascal 3.6

Изложенный ниже метод применяется ко всем файлам, содержащим исходный текст, и входящим в состав программной документации по отдельности. Для каждого файла вносится в протокол запись: «Требования к исходным кодам в части компиляции во Free Pascal 3.6 для файла #####» – соответствует/не соответствует (где вместо ##### указывается название файла).

Проверка осуществляется путем открытия файла с исходным кодом в среде Free Pascal 3.6, осуществления компиляции (обе работы делаются в соответствии с документацией к Free Pascal). В случае, если компиляция завершилась успешно в протокол заносится запись о соответствии требованиям, в противном случае – о несоответствии.



### 2.2.3 Пояснительная записка технического проекта

Технический проект – это модель будущего программного изделия (или его частей) и будущей разработки. На этом этапе исполнитель (в больших проектах) проводит исследование с целью:

- выявления наиболее оптимального способа будущей разработки программного продукта (он выбирает языки программирования, среды разработки, другие необходимые программные и аппаратные средства);
- изучения научной и/или технической литературы с целью получения информации о том, как в настоящее время осуществляется решение тех или иных подзадач, что встречаются внутри данной задачи (например, технический проект может в себя включать изучение математических основ нейронных сетей и их адаптацию для конкретной задачи): результатом этого этапа могут быть изложение алгоритмов, которые будут использованы в будущем программном продукте;
- построения модели будущего программного продукта с тем или иным уровнем детализации (выделение основных частей, их характеристик, способов взаимодействия между ними);
- в случае обмена информацией с другими программными изделиями и/или использования файлов, в технический проект включается информация о выборе формата хранения/передачи данных.

В проектах того масштаба, что мы рассматриваем, стадия технического проекта может быть опущена. Однако, мы ее включаем в состав жизненного цикла с целью формального документирования процесса выбора архитектурных решений и конкретных алгоритмов.

Разрабатывая технический проект в самостоятельной работе следует учитывать, что программа должна состоять из собственно программы и модуля, осуществляющего основные вычисления. Цель такого разделения состоит в том, что перед написанием программы и модуля необходимо будет создать другую программу, которая предназначена для автоматического тестирования этого модуля.

Писать программу автоматического тестирования и модуль в самостоятельной работе будут разные студенты для одного и того же варианта, однако способ использования модуля (то есть его интерфейс) должен быть фиксирован до начала написания как основной программы, так и программы автоматического тестирования.

Собственно целью технического проекта в самостоятельной работе будет:

- выбор метода решения задачи;
- выбор алгоритма решения задачи и документирование этого выбора, в том числе, путем рисования блок-схемы.
- выбор интерфейса модуля и документирование этого выбора;

Начальная часть документа похожа на начальные части технического задания и программы и методики испытаний.

## 1. Введение

### 1.1 Наименование программы

Решение квадратного уравнения

### 1.2 Условное обозначение темы разработки

ПП-00

### 1.3 Документ, на основании которого ведется разработка

Сборник самостоятельных работ по МДК01.02 Прикладное программирование

## 2. Назначение и область применения

### 2.1 Назначение программы

Программа предназначена для автоматического решения квадратных уравнений

### 2.2 Краткая характеристика области применения

Программа может использоваться любыми лицами, деятельность которых требует решения конкретных квадратных уравнений.

## 3. Технические характеристики

### 3.1 Постановка задачи на разработку программы

Программа предназначена для решения конкретных квадратных уравнений вида  $ax^2 + bx + c = 0$ , где  $a$ ,  $b$  и  $c$  – заданные вещественные числа.

### 3.2 Описание применяемых математических методов

Применяемый метод имеет ограничение:  $a \neq 0$ .

Для решения используются стандартные формулы с использованием дискриминанта.

1. Вычисляется дискриминант по формуле:  $D = b^2 - 4ac$ .

2. Если  $D < 0$ , то уравнение не имеет решения

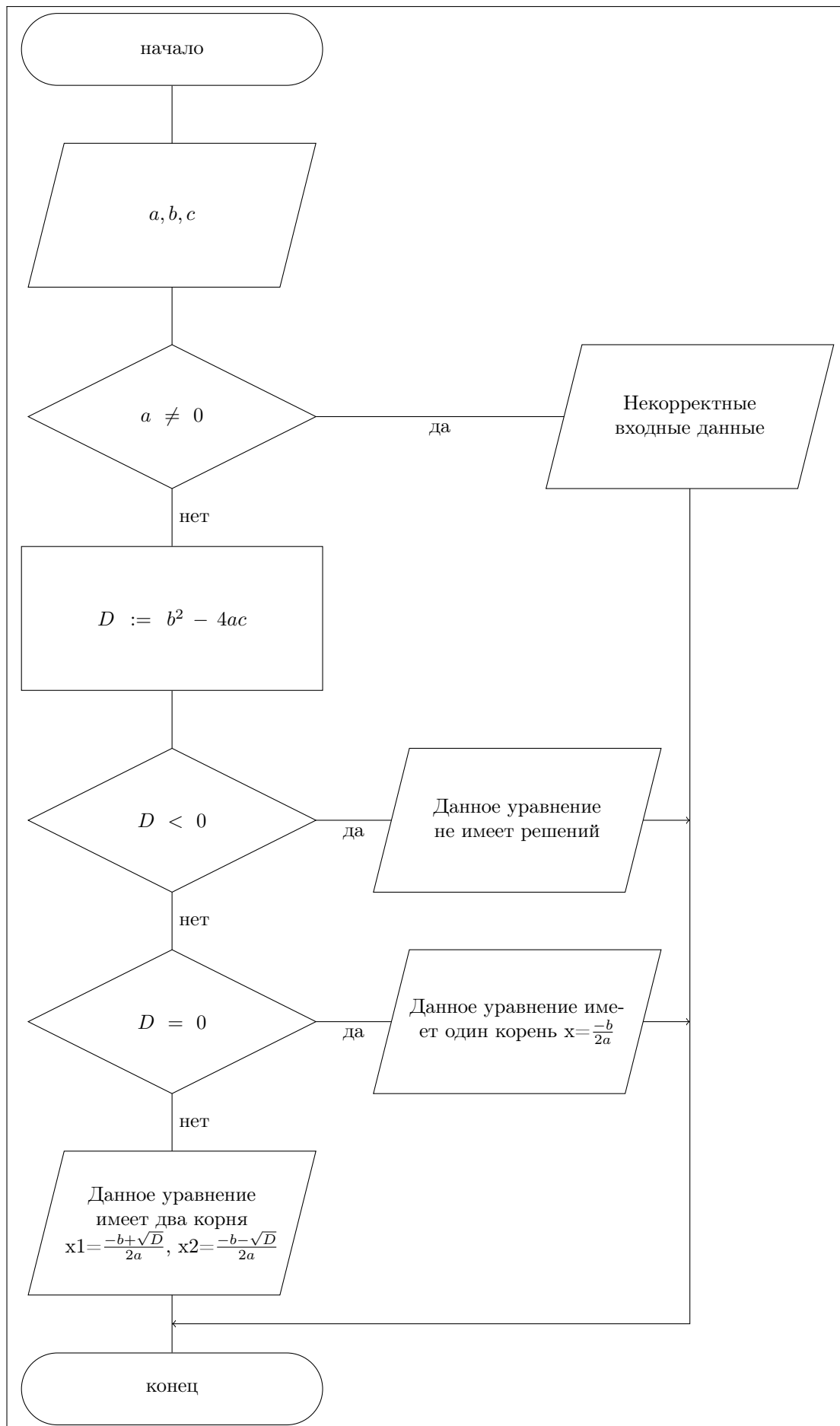
3. Если  $D = 0$ , то уравнение имеет один корень:  $x = \frac{-b}{2a}$ .

4. Если  $D > 0$ , то уравнение имеет два корня:  $x_1 = \frac{-b + \sqrt{D}}{2a}$ ,  $x_2 = \frac{-b - \sqrt{D}}{2a}$ .

Метод выбран в соответствии с п. 4.5.2 технического задания.

### 3.3 Описание алгоритма

Выбранный метод решения непосредственно приводит к следующему алгоритму:



3.4 Описание и обоснование выбора метода организации входных и выходных данных  
В разрабатываемом программном изделии выбор метода организации входных и выходных данных не требуется

3.5 Описание и обоснование выбора состава технических и программных средств  
Состав технических и программных средств полностью описан в техническом задании (пп. 4.4; 4.5.4; 4.5.5)

3.6 Интерфейс программного модуля

Вычислительную часть программы содержит модуль `solveseq`, содержащий функцию `solve`:

```
tsolve=(bad_data,roots_0,roots_1,roots_2);
```

```
function solve (a,b,c: real; var x1,x2: real): tsolve;
```

Функции передаются аргументы  $a$ ,  $b$  и  $c$  – коэффициенты квадратного уравнения  $ax^2 + bx + c = 0$ .

Возвращаемые значения:

<code>bad_data</code> ,	если $a = 0$
<code>roots_0</code> ,	если уравнение не имеет корней
<code>roots_1</code> ,	если уравнение имеет один корень (он передается в параметре $x_1$ )
<code>roots_2</code> ,	если уравнение имеет два корня (они возвращаются в параметрах $x_1$ и $x_2$ )

4. Источники, использованные при разработке

Квадратное уравнение [Электронный ресурс] – Режим доступа:

[https://ru.wikipedia.org/wiki/Квадратное\\_уравнение](https://ru.wikipedia.org/wiki/Квадратное_уравнение)

### 2.2.4 Программа автоматического тестирования модуля

Программа автоматического тестирования модуля должна разрабатываться на основе пояснительной записки технического проекта, в котором описан интерфейс будущего модуля. Используя этот интерфейс, студент должен запустить на достаточном числе примеров подпрограммы, входящие в модуль, и проверить, что они возвращают правильные результаты.

В случае, если модуль осуществляет работу с файлами, программа автоматического тестирования автоматически создает файлы с тестовыми данными, передает их название в подпрограмму модуля и проверяет корректность возвращенных данных (или содержимое результирующего файла).

По окончании работы программа автоматического тестирования модуля выводит на экран либо предложение «Все тесты успешно пройдены», либо предложение «Тест *такой-то* прошел с ошибкой».

Программа сдается на проверку в электронном виде, а также в виде распечатки.

### 2.2.5 Программа

На этом этапе студент разрабатывает программу в соответствии с техническим заданием и пояснительной запиской технического проекта. При этом он проверяет, что программа автоматического тестирования модуля успешно выполняется при подключении созданного им модуля.

Программа сдается на проверку в электронном виде, а также в виде распечатки.

### 2.2.6 Протокол испытаний

На этом этапе студент разрабатывает документ, который в начальной своей части похож на программу и методику испытаний (от титульной страницы до пункта 5.2 включительно), а далее студент вставляет протокол в форме, разработанной самостоятельно.

## 2.3 Варианты заданий

Замечание: объем программы и тип используемых конструкций языка достаточно сильно отличаются, так как написание программы не главное в самостоятельной работе. Предполагается, что все предлагаемые задания просты. Некоторые задания специально сформулированы недостаточно точно, уточнение производит студент при разработке технического задания.

1. По данному файлу создать второй файл, в котором все байты записаны в обратном порядке
2. В данном текстовом файле определить количество строк
3. В данном текстовом файле определить количество слов (слово – непрерывная последовательность символов, состоящая из букв и знаков дефиса, слово может продолжаться на следующей строке (перенос))
4. В данном текстовом файле определите статистику использования разных символов.

5. По данному текстовому файлу создайте новый текстовый файл, содержащий в начале каждой строки ее номер.
6. По данному вещественному числу определить как оно представляется внутри компьютера, а также разбить это представление на участки: мантисса, порядок, знак (эта программа должна будет правильно работать только под фиксированным транслятором языка Паскаль).
7. По данной строке определить сколько строк можно получить переставляя символы: для строки «мама» ответ – 6: «ааммм», «амам», «амма», «ммаа», «мама», «маам» (сами варианты перебирать не надо). Подсказка: формулу можно найти на странице по ссылке: [http://www.chem-astu.ru/chair/study/probability-theory/1\\_Combinatorics.shtml](http://www.chem-astu.ru/chair/study/probability-theory/1_Combinatorics.shtml)
8. По данным координатам точек определить, образуют ли они как вершины выпуклый многоугольник (координаты точек вводятся в том же порядке, в каком они будут в многоугольнике).
9. Нахождение суммы ряда:  $1 + \frac{x^1}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots$ , где  $x$  – заданное **комплексное** число.
10. Программа создает «архив», в котором записаны все файлы данной папки без сжатия с указанием их названия. Указание: изучите процедуры `findFirst` и `findNext`.
11. В текстовом файле найти сумму неотрицательных целых чисел, стоящих в начале строк.
12. Сравнение двух файлов на предмет их полного совпадения.
13. Сравнение двух текстовых файлов на предмет их совпадения за исключением возможного наличия «лишних» пробелов в концах строк.
14. В данном текстовом файле найти наиболее часто используемое слово (включая переносы: то есть когда на одной строке начинается слово и идет дефис, а на другой стороне продолжается). Случай, когда само слово содержит дефис может обрабатываться некорректно.
15. Решение уравнения вида  $ax^2 + bx + c = 0$  с комплексными коэффициентами (и в множестве комплексных чисел).
16. Разложение данного натурального числа в произведение простых сомножителей (результат как в школе).
17. Сравнение двух текстовых файлов на предмет их совпадения за исключением того, что количество пробелов между другими символами может различаться.
18. В данном текстовом файле, содержащем программу на языке Паскаль, проверить правильность размещения комментариев (все комментарии закрыты, правильно вложены и т. п.).
19. Проверить, является ли данный файл текстовым файлом (содержащим только текст) /расширение файла может быть любым/.
20. Проверить, совпадают ли два заданных текстовых файла за исключением того, что между строками файла могут вставляться пустые строки.
21. Решить систему уравнений:
$$\begin{cases} ax + b = 0 \\ cx + d = 0 \end{cases},$$
предполагая, что коэффициенты и неизвестные – комплексные числа.
22. По координатам двух точек на евклидовой плоскости нахождение длины отрезка, концы которого – заданные точки.
23. Проверка простоты заданного целого числа.
24. Нахождение числа Фибоначчи по номеру.
25. Нахождение номера числа Фибоначчи по его значению.
26. Нахождение площади треугольника по длинам его сторон.
27. Нахождение стороны треугольника по двум другим сторонам и углу между ними.
28. Нахождение евклидовых координат по полярным координатам точки.
29. Нахождение отношения двух комплексных чисел.

30. Нахождение стороны по другой стороне и двум, прилежащим ей углам.
31. Нахождение суммы цифр в целом числе.
32. Нахождение суммы двух обыкновенных дробей (результат – тоже обыкновенная дробь).
33. Проверка того, что заданное целое число совершенное (то есть равно всех его делителей, кроме себя).
34. Проверка того, что заданная точка в евклидовой плоскости лежит в круге с заданным центром и радиусом. Точки задаются координатами.
35. Нахождение наибольшего общего делителя двух данных целых чисел.
36. По заданному целому числу нахождение количества его целых делителей.
37. По данным трем числам определение корректности даты (первое число – номер дня в месяце, второе – месяц, третье – год).
38. Перевод целого числа из 10-ой в двоичную систему счисления.
39. Перевод целого числа из двоичной в десятичную систему счисления.
40. Возведение комплексного числа в натуральную степень.
41. Решение системы двух линейных уравнений с двумя неизвестными.
42. Решение линейного уравнения с комплексными коэффициентами.
43. Нахождение обратной матрицы для данной матрицы  $2 \times 2$ .
44. Вывести данное натуральное число словами на русском языке (по числу 121 – «сто двадцать один»).
45. Для данных чисел нахождение среднего арифметического  $S$ , среднего арифметического квадратов –  $S_2$  и среднеквадратического отклонения –  $\sqrt{\frac{n}{n-1}(S_2 - S^2)}$ .
46. Нахождение всех способов представления данного натурального числа в виде суммы квадратов натуральных чисел.
47. Нахождение площади треугольника по координатам вершин.
48. Нахождение натуральной степени матрицы  $2 \times 2$ .
49. Нахождение наименьшего общего кратного двух заданных целых чисел.
50. Проверка того, что заданная точка в евклидовой плоскости лежит внутри треугольника, заданного координатами вершин (все координаты – целые).
51. Расчет количества денег вклада по начальной сумме, годовому проценту и количеству лет (предполагается ежегодная капитализация процентов).
52. Для данной обыкновенной дроби, заданной числителем и знаменателем, найти равную ей несократимую правильную дробь.
53. Проверка пары чисел на взаимную простоту.
54. Нахождение суммы ряда:  $1 + \frac{x^1}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots$ , где  $x$  – заданное число.
55. Вывести символами псевдографики прямоугольник на экран по координатам расположения верхнего левого и правого нижнего угла. Прямоугольник имеет стороны параллельные краям экрана.
56. Дана строка, содержащая только буквы латинского алфавита и пробелы. Осуществить простейшее шифрование строки, поменяв символы на символ, стоящий в алфавите через  $n$  символов правее (если при этом произойдет выход за пределы алфавита, то необходимо перейти к его началу). Пример: для  $n = 2$  и строки «Hello world» ответ: «Jgnnq yqtnf».
57. По координатам позиции двух ферзей на шахматной доске определить, бьют ли они друг друга.
58. По данным трем числам (день-месяц-год) нахождение дату следующего дня (день-месяц-год)
59. По данному числу определить, сколько раз встречается цифра 1 в его двоичной записи.
60. Перевод комплексного числа из обычной формы в показательную (или тригонометрическую – по выбору студента).

### 3 Самостоятельная работа №2. Указатели и списки (7 часов)

#### Цели работы

- Практическое закрепление методики работы с простейшими списками
- Подготовка к закреплению понятия полиморфизма в объектно-ориентированном программировании
- проработка профессиональных компетенций ПК1.2-1.5
- проработка общих компетенций ОК2, 3, 5, 6.

#### Общая часть задания

1. Что выведет следующая программа на экран?

```
var p,p2: ^integer;  
    a,b: integer;  
begin  
  a:=2;  
  p:=@a;  
  b:=3;  
  p2:=p;  
  p:=@b;  
  p2^:=p^*10+p2^;  
  writeln (a, ' ',b);  
end.
```

2. Что выведет следующая программа на экран?

```
type pt=^t;  
t=record  
  e: integer;  
  n: pt;  
end;  
var  
p: pt;  
x: integer;  
begin  
  new (p);  
  p^.e:=3;  
  new (p^.n);  
  p^.n^.n:=nil;  
  p^.n^.e:=4;  
  x:=0;  
  while (p<>nil) do  
  begin  
    x:=x*10+p^.e;  
    p:=p^.n;  
  end;  
  writeln (x);  
end.
```

3. Что выведет следующая программа на экран?

```
type pt=^t;  
  t=record  
    e: integer;  
    n: pt;  
  end;  
var  
p,p2,p3: pt;  
i: integer;  
begin
```

```

new (p);
p^.e:=1;
p^.n:=nil;
p3:=p;
for i:=2 to 3 do begin
  new (p2);
  p2^.e:=p^.e*3;
  p2^.n:=nil;
  p^.n:=p2;
  p:=p2;
end;
while (p3<>nil) do begin
  writeln (p3^.e);
  p3:=p3^.n;
end;
end.

```

4. Напишите фрагмент программы, который добавляет новый элемент в конец следующего списка:

```

type PItem=^TItem;
  TItem=record
    el: integer;
    next: PItem;
  end;
var first: PItem;

```

5. Напишите фрагмент программы, который удаляет первый элемент из списка:

```

type PItem=^TItem;
  TItem=record
    el: integer;
    next: PItem;
  end;
var first: PItem;

```

В соответствии со своим вариантом выполните действия со списком, хранящимся с использованием следующих типов и переменных

```

type PItem=^TItem;
  TItem=record
    el: integer;
    next: PItem;
  end;
var first: PItem;

```

**Просмотр списка должен осуществляться ровно один раз.**

1. Нахождение суммы всех элементов исходного списка, удаление всех элементов, равных 0, добавление квадрата элемента после каждого элемента списка, полученного после завершения предыдущих преобразований.
2. Нахождение произведения всех элементов исходного списка, удаление всех элементов, меньших 0, добавление единицы после каждого четного элемента списка, полученного после завершения предыдущих преобразований.
3. Нахождение максимума всех элементов исходного списка, удаление всех четных элементов, добавление копии элемента после каждого элемента списка, пропорционального 3.
4. Нахождение минимума всех элементов исходного списка, удаление всех нечетных элементов, добавление после каждого элемента исходного списка его номера (каким он был в исходном списке).
5. Нахождение суммы всех четных элементов исходного списка, удаление всех чисел, оканчивающихся на 0, из списка, добавление после каждого элемента списка квадрата этого элемента.



6. Нахождение максимума всех четных элементов исходного списка, удаление всех нечетных элементов и добавление после каждого элемента списка, являющегося квадратом целого числа, значения  $-1$ .
7. Нахождение минимума всех элементов, больших нуля; удаление всех отрицательных элементов и добавление после каждого элемента списка числа, на единицу большего.
8. Нахождение суммы всех элементов; удаление всех элементов, пропорциональных двум, добавление после каждого элемента списка куба этого элемента.
9. Нахождение суммы всех элементов, меньших нуля; удаление всех элементов, последняя цифра которых равна 5, добавление после каждого элемента списка числа, равного последней цифре данного элемента.
10. Нахождение наименьшего четного числа; удаление всех элементов, больших 10, добавление после каждого элемента списка квадрата данного числа.
11. Нахождение наименьшего числа, кратного трем; удаление всех элементов – двузначных натуральных чисел, добавление после каждого элемента, пропорционального двум, копии этого числа.
12. Нахождение наибольшего номера элемента списка (в исходном списке) числа, кратного трем; удаление всех элементов, в которых последние две цифры совпадают, добавление после каждого элемента числа, на единицу меньшего.
13. Нахождение суммы чисел, кратных трем; удаление всех элементов, оканчивающихся на 5, добавление после каждого положительного элемента, копии этого числа.
14. Нахождение произведения однозначных натуральных чисел; удаление всех элементов, меньших по модулю 5, добавление после каждого отрицательного элемента, копии этого числа.
15. Нахождение наибольшего номера элемента списка (в исходном списке) двузначного числа; удаление всех элементов, в которых последняя цифра больше 5, добавление после каждого элемента числа, на единицу большего.
16. Нахождение суммы всех чисел, кратных 5; удаление элементов, являющихся квадратом натурального числа, добавление после каждого положительного элемента его копии.
17. Нахождение произведения всех чисел, кратных 5; удаление элементов, стоящих на позиции, номер которой (в исходном списке) совпадает с самим элементом, добавление после каждого элемента его копии, уменьшенной в 10 раз (целой части полученного частного).
18. Нахождение наименьшего числа, кратного десяти; удаление всех двузначных натуральных чисел, добавление после трехзначных натуральных чисел копии.
19. Нахождение наибольшего числа, кратного десяти; удаление всех чисел, в которых разряд десятков и единиц совпадает, добавление после каждого числа, пропорционального последней цифре частного от деления числа на последнюю цифру.
20. Нахождение количества чисел, у которых последняя цифра  $-5$ ; удаление всех чисел, стоящих после числа, на единицу большего (в исходном списке), добавление после положительных чисел элемента списка, равного единице.
21. Нахождение произведения однозначных чисел списка; удаление чисел, меньших предыдущего (в исходном списке), добавление после каждого элемента остатка от деления его на 3.
22. Нахождение суммы всех четных чисел списка; удаление чисел, в которых разряд сотен равен разряду единиц, вставление после каждого элемента списка суммы его и первого элемента списка.
23. Удаление элементов равных номеру элемента в исходном списке, добавление после каждого элемента числа его единиц, нахождение минимума из оставшихся элементов списка.
24. Удаление элементов, пропорциональных последней цифры, добавление после каждого элемента его номера (каким бы он был после удаления элементов), нахождение максимума из оставшихся элементов списка.
25. Удаление элементов – однозначных чисел, добавление после каждого элемента его последней цифры, нахождение суммы оставшихся элементов списка.
26. Нахождение суммы всех положительных элементов исходного списка, удаление всех элементов, меньших 0, добавление куба элемента после каждого элемента списка, полученного после завершения предыдущих преобразований.

27. Нахождение произведения всех положительных элементов исходного списка, удаление всех элементов, больших 0, добавление на единицу большего элемента после каждого четного элемента списка, полученного после завершения предыдущих преобразований.
28. Нахождение максимума всех отрицательных элементов исходного списка, удаление всех нечетных элементов, добавление числа, на единицу меньшего, чем элемент списка, после каждого элемента списка.
29. Нахождение минимума всех положительных элементов исходного списка, удаление всех четных элементов, добавление после каждого элемента исходного списка его номера (каким он был в исходном списке).
30. Нахождение суммы всех четных элементов исходного списка, удаление всех чисел, оканчивающихся на 5, из списка, добавление перед каждым элементом списка его копии.
31. Нахождение максимума всех нечетных элементов исходного списка, удаление всех четных элементов и добавление после каждого элемента списка, являющегося кубом целого числа, значения  $-1$ .

## 4 Самостоятельная работа №3. ООП (6 часов)

### Цели работы

- Практическое закрепление знания принципов объектно-ориентированного программирования
- проработка профессиональных компетенций ПК1.2
- проработка общих компетенций ОК2, 3, 5, 6.

Напишите программу осуществляющую ввод информации о сущностях, описанных в вашем варианте задания и вывод на экран некоторых из них. Количество вводимых сущностей не ограничено (нужно использовать списки); обязательно использовать ООП, инкапсуляцию, наследование и полиморфизм. Проверять корректность входных данных и делать проверку того, что хватает памяти не обязательно. Необходимо корректно освободить память от объектов и списка.

1. Товары Интернет-магазина – книги и диски. Все товары определяются ценой, книги имеют название, автора, количество страниц; диски – название, количество треков. Выведите на экран все товары со стоимостью меньше 100 рублей.
2. Преподаватели определяются ФИО. Для тех, кто имеют диссертацию дополнительно вводится ее название; для остальных – стаж работы. Вывести всех преподавателей, у которых ФИО начинается на букву «А».
3. Телефоны определяются названием модели. Проводные телефоны дополнительно определяются типом номеронабирателя (диск или кнопки); а беспроводные – дальностью действия радиосигнала. Вывести все телефоны, название которых начинается на «А».
4. Покатушки определяются названием и расстоянием. Однодневные катушки дополнительно определяются плановым временем поездки (в часах). Многодневные катушки определяются количеством дней и категорией сложности похода (от 1 до 6). Вывести все покатушки длиной более 100 км.
5. Музыкальная композиция определяется названием и композитором. Дополнительно для песни указывается автор стихов. Выведите информацию о всех композициях, у которых композитор начинается на букву «А».
6. Олимпиада определяется названием. Если олимпиада участвует в программе приема в ВУЗы дополнительно указывается уровень олимпиады (1–3), если олимпиада – этап всероссийской, то указывается название этапа (школьная, окружная, региональная, всероссийская), в остальных случаях – размер призового фонда. Выведите все олимпиады, название которых начинается на букву «А».
7. Проездной билет определяется стоимостью. Билет на количество поездок определяется количеством поездок. Билет на неограниченное количество поездок определяется сроком действия (1 день, 5 дней, 10 дней, 15 дней, месяц, три месяца, 6 месяцев, год). Выведите информацию о билетах, стоимостью меньше 300 рублей.
8. Информация о студенте определяется ФИО. Для студентов, не имеющих автомата, указывается балл, полученный на экзамене (2–5); для студентов, имеющих автомат указывается основание (олимпиада или контрольные работы). В случае, если контрольная работа – то также указывается средний балл за к/р. Выведите всю информацию о студентах с фамилией, начинающейся на буквы от А до К.

9. Сотовый телефон определяется названием. Для смартфонов указывается операционная система. А для других телефонов – наличие браузера. Выведите информацию о телефонах, название которых содержит слово «Nokia».
10. Куртка определяется названием модели, наличием капюшона. Для мембранных курток указывается степень водонепроницаемости (число в мм рт. ст.), для остальных – наличием пропитки. Выведите информацию обо всех куртках, имеющих капюшон.
11. Жесткий диск определяется названием и емкостью. Внешние жесткие диски определяются дополнительно наличием системы, смягчающей последствия падения. Внутренние жесткие диски – размером (2.5/3.5 дюйма). Выведите информацию о дисках, емкостью больше 200 Гб.
12. Велосипед определяется названием модели. Горному велосипеду соответствует количество скоростей, ВМХ – тип конструкции (фривил, кассетная, фрикостер). Выведите информацию обо всех велосипедах, содержащих в названии «Nogso».
13. Электронная книга определяется названием и размером экрана. Для EInk-дисплея указывается поколение (pearl, vizplex); для LCD – количество поддерживаемых цветов. Выведите информацию о всех книгах с размером экрана не менее 7 дюймов.
14. GPS определяется названием, диагональю экрана. Для переносных GPS указывается наличие велосипедного крепления; для автомобильных – поддержка отображения пробок и наличие радар-детектора. Выведите информацию обо всех GPS с размером экрана менее 7 дюймов.
15. Пылесос определяется названием модели. Для обычного пылесоса указывается мощность, для пылесоса-робота – размер убираемого помещения и количество виртуальных стен. Выведите информацию обо всех пылесосах, содержащих в названии слово Indesit.
16. Туры определяются названием. Для пляжного тура указывается тип пляжа (галечный, песок); для экскурсионного – количеством экскурсий. Выведите информацию обо всех турах, содержащих слово Египет.
17. Язык программирования определяется названием. Алгоритмические языки определяются поддержкой ООП (отсутствует, на классах, прототипная), остальные языки – типом (функциональный, логический, стиль ). Выведите информацию обо всех языках, название которых начинается с буквы «А».
18. Контрагенты определяются названием. Индивидуальные предприниматели дополнительно определяются наличием счета в банка, а юридические лица – формой организации (ООО, ОАО, ЗАО). Выведите информацию обо всех контрагентах, название которых начинается с буквы «А».
19. Счет в банке определяется номером. Для текущего счета указывается плата за обслуживание, для сберегательного счета – проценты годовых и наличие капитализации. Выведите информацию обо всех счетах, номер которого начинается с 408178...
20. Автомобильная дорога определяется названием и километражом. Бесплатная дорога определяется статусом автомагистрали (автомагистраль или нет), а платная – стоимостью за километр для обычных пользователей. Выведите информацию о дорогах, длина которых менее 100 км.
21. Офисное здание определяется адресом. В случае наличия стоянки указывается количество машиномест и стоимость аренды за месяц. Выведите информацию о зданиях, в адресе которых присутствует слово Тверская.
22. Товары Интернет-магазина – GPS-навигаторы и карты. Все товары определяются ценой и названием, GPS-навигаторы имеют назначение (ручной, автомобильный) и признак возможности загрузки карт; карты – размером (в Мб). Выведите информацию о всех товарах со стоимостью менее 4000 рублей.
23. Товары Интернет-магазина – чай и кофе. Все товары определяются ценой, названием и весом, кофе – типом (растворимый, молотый, в зернах), чай – типом (черный, зеленый). Выведите информацию о всех товарах с весом менее 150 г.
24. Объекты продаваемые в коттеджном поселке: участки (определяются площадью, стоимостью, наличием подряда), дома (определяются этажностью, площадью и стоимостью). Выведите все объекты со стоимостью меньше 1000000 рублей.
25. Вопросам теста соответствует формулировка и количество баллов за правильный ответ. Вопросам с вариантами правильных ответов соответствует 4 варианта ответа и номер правильного ответа; остальным вопросам – формулировка правильного ответа. Выведите все вопросы, оцениваемые в 10 баллов и выше.

26. Слова определяются собственно словом. Для существительных указывается род, для глаголов – спряжение. Выведите информацию обо всех словах, начинающихся на букву «А».
27. Операционная система определяется названием. Для операционной системы на базе Linux указывается название менеджера пакетов; для остальных – стоимость лицензии. Вывести все операционные системы, у которых название начинается на букву «А».
28. Рюкзаки определяются названием модели и емкостью. Для городских рюкзаков указывается наличие «вентилируемой спины» для походных – количество отделений и наличие крепления для трекинговых палок. Вывести информацию обо всех рюкзаках, в названии которых присутствует слово «Трек».
29. Автостоянка определяется названием, количеством машиномест. Для крытой автостоянки указывается количество этажей. Для открытой стоянки – наличие охраны. Вывести информацию обо всех автостоянках с количеством мест больше 20.
30. Партия определяется названием. Для тех партий, что финансируются из бюджета указывается размер ассигнований, а для остальных – количество депутатов в каких-либо представительных органах власти. Вывести информацию обо всех партиях с названиями, начинающимися на буквы от «А» до «К».
31. Специальность определяется названием и кодом. Для специальности высшего образования дополнительно указывается тип (специалитет/бакалавриат/магистратура), а для специальности среднего – длительность обучения в месяцах. Вывести информацию о специальностях с кодом, начинающихся на «09».

## 5 Самостоятельная работа №4. Конспект (3 часа)

### Цели самостоятельной работы

- ознакомление с свойствами и событиями на примере двух компонент;
- развитие навыков поиска, анализа и оценки информации;
- проработка общих компетенций ОК2, 4, 5, 7

Кратко письменно (от руки) опишите все доступные для программиста через инспектор объектов свойства и события для TForm и TEdit.

## 6 Самостоятельная работа №5. Видеоурок «простейшие компоненты» (3 часа)

Видеоурок в данной самостоятельной работе должен представлять практическое введение в использование соответствующих компонентов в собственных программах. Каждую тему готовят по два студента, при этом в озвучке должны участвовать оба участника.

Каждый видеоурок заслушивается аудиторией, после чего студенты и/или преподаватель могут задавать дополнительные вопросы. Во время ответов на вопросы предыдущей группы очередная группа студентов готовит оборудование.

Слушатели доклада конспектируют основные его моменты (содержание докладов входит в зачет и в опросы на занятиях).

### Цели самостоятельной работы

- ознакомление с простейшими компонентами;
- развитие навыков организации собственной деятельности;
- развитие навыков поиска, анализа и оценки информации;
- развитие навыков работы в команде.
- проработка общих компетенций ОК2-7

В случае, если студент не выступал на занятиях, где заслушивались доклады, то он впоследствии (до зачета) лично сдает в форме устного опроса (вне занятий) содержание всех тем докладов.

Темы видеоуроков следует выбирать строго в порядке их включения в данное пособие.

1. TMemo;
2. TMaskEdit;
3. TSpinEdit;
4. TFloatSpinEdit;
5. TCheckBox;
6. TRadioButton;
7. TTimer;
8. TProgressBar;
9. TShape;
10. TOpenDialog;
11. TSaveDialog;
12. TSelectDirectoryDialog;
13. TFindDialog;
14. TReplaceDialog;
15. TCalendarDialog;
16. TColorDialog;
17. TFontDialog;
18. TTrayIcon;
19. TOpenPictureDialog;
20. TSavePictureDialog;
21. TCalculatorDialog;
22. TCalculator.

## **7 Самостоятельная работа №6. Доклады и видеоуроки: простейшие компоненты, работающие с графической информацией (3 часа)**

### **Цели самостоятельной работы**

- ознакомление с компонентами, работающими с графикой;
- развитие навыков организации собственной деятельности;
- развитие навыков поиска, анализа и оценки информации;
- развитие навыков работы в команде.
- проработка общих компетенций ОК2-7

1. Доклад. Подробный обзор компонента TImage. Краткий обзор TImageList (2 человека)
2. Доклад. Подробный обзор рисования с использованием свойства Canvas компонента TImage (4 человека)
3. Видеоурок LineSeries (2 человека)
4. Видеоурок PolarSeries (2 человека)
5. Видеоурок AreaSeries (2 человека)
6. Видеоурок BarSeries (2 человека)

7. Видеоурок PieSeries (2 человека)
8. Видеоурок ParametricCurveSeries (2 человека)
9. Видеоурок User-drawn Series (2 человека)
10. Видеоурок FunctionSeries (2 человека)
11. Видеоурок+доклад Least-squares fit series (4 человека)
12. Видеоурок+доклад B-SplineSeries (4 человека)
13. Видеоурок+доклад CubicSplineSeries (4 человека)
14. Видеоурок Color Map Series (2 человека)

## 8 Самостоятельная работа №7. Доклад и видеоурок: компоненты для работы с большими наборами данных (3 часа)

### Цели самостоятельной работы

- ознакомление с компонентами, работающими с различными списками;
- развитие навыков организации собственной деятельности;
- развитие навыков поиска, анализа и оценки информации;
- развитие навыков работы в команде.
- проработка общих компетенций ОК2-7

Каждую тему готовят по 4 человека.

1. TListBox
2. TComboBox
3. TStringGrid
4. TDrawGrid
5. TTreeView
6. TListView
7. TValueListEditor
8. TApplicationProperties

## 9 Самостоятельная работа №8. Паттерны проектирования (3 часа)

### Цели самостоятельной работы

- ознакомление с основными паттернами (шаблонами) проектирования в объектно-ориентированной парадигме;
- развитие навыков организации собственной деятельности;
- развитие навыков поиска, анализа и оценки информации;
- развитие навыков работы в команде.
- проработка общих компетенций ОК2-7, 9

## Формат проведения

Доклады готовятся группами по 2 человека. В виде исключения доклад может готовиться тремя студентами. На занятия, где будут заслушиваться доклады, необходимо принести: презентацию, печатную версию доклада, раздаточный материал (по желанию).

Каждый доклад заслушивается аудиторией, после чего студенты и/или преподаватель задает дополнительные вопросы. Во время ответов на вопросы предыдущей группы очередная группа студентов готовит оборудование.

Слушатели доклада конспектируют основные его моменты (содержание докладов входит в зачет и в опросы на занятиях).

## Параметры, по которым оценивается самостоятельная работа

- качество доклада
  - раскрытие темы;
  - наличие презентации;
  - качество ответов на дополнительные вопросы преподавателя и аудитории;
  - общая композиция (вступление – основная часть – выводы);
  - качество выступления (отсутствие чтения «по бумажке», грамотность речи, понятность, точность);
  - соблюдение регламента (7 минут выступления – можно меньше, 5 минут ответы на вопросы, 2 минуты – перерыв между докладами) – это требование обозначает, что надо выделить совсем главное и кратко его изложить, также требуется определенная репетиция;
- качество презентации
  - небольшое количество слайдов (примерно один слайд на минуту);
  - на слайдах представлены либо основные тезисы либо иллюстрирующие доклад материалы (ни в коем случае нельзя писать в них весь текст доклада или даже его значительную часть, текст должен быть виден всей аудитории);
  - отсутствие «изысков», отвлекающих внимание от главного;
- содержание
  - зачем нужны паттерны темы доклада (какую проблему они решают, цель их использования – это главное);
  - примеры возможного их использования (на уровне концепции);
  - пример кода (не в деталях, а в основных моментах) /диаграмма UML с пояснениями/;
  - пример использования в C++ (STL), если есть;
- активность как слушателя:
  - наличие вопросов;
  - возможность ответить по только что заслушанному докладу;
- качество печатного варианта доклада:
  - полнота;
  - оформление;
  - стиль изложения.

В случае, если студент не выступал на занятиях, где заслушивались доклады, то он впоследствии (до зачета) лично сдает в форме устного опроса (вне занятий) содержание всех тем докладов.

## Темы докладов

Во избежании путаницы приводятся английские названия паттернов. Список приводится в порядке убывания приоритета.

1. MVC
2. Iterator, Composite
3. Chain of Responsibility, Observer
4. Factory, Factory Method, Abstract Factory
5. Singleton, Flyweight, Lazy Initialization
6. Strategy, Visitor,
7. Adapter (Wrapper, Translator), Bridge,
8. Decorator, Facade, Proxy
9. Builder, Template Method
10. Command, Functor
11. Memento, Prototype
12. Multiton, Object Pool
13. Null Object, Private Class Data, Mediator

## 10 Самостоятельная работа №9. Эссе (1 час)

### Цели самостоятельной работы

- проектирование собственного профессионального будущего;
- проработка общих компетенций ОК1, 4, 8

Дайте развернутый и обоснованный ответ на вопрос: «Что я хочу изучить в будущем в профессиональной сфере?»

## 11 Список литературы

1. ГОСТ 19.001-77 «Единая система программной документации. Общие положения»
2. ГОСТ 19.102-77 Единая система программной документации: стадии разработки
3. ГОСТ 19.101-77 «Единая система программной документации. Виды программ и программных документов»
4. ГОСТ 19.301-79\* «Единая система программной документации. Программа и методика испытаний. Требования к содержанию и оформлению»
5. ГОСТ 19.201-78 «Единая система программной документации. Техническое задание. Требования к содержанию и оформлению»
6. ГОСТ 19.701-90 «Единая система программной документации. СХЕМЫ АЛГОРИТМОВ, ПРОГРАММ, ДАННЫХ И СИСТЕМ. Обозначения условные и правила выполнения»
7. ГОСТ 19.404-79 «Единая система программной документации. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА. ТРЕБОВАНИЯ К СОДЕРЖАНИЮ И ОФОРМЛЕНИЮ»
8. <http://www.oodesign.com/>
9. [http://en.wikipedia.org/wiki/Design\\_pattern\\_\(computer\\_science\)](http://en.wikipedia.org/wiki/Design_pattern_(computer_science))
10. <http://citforum.ru/SE/project/pattern/>
11. Э.Гамма, Р. Хелм, Р. Джонсон, Дж. Влссидес «Приемы объектно-ориентированного проектирования. Паттерны проектирования» – СПб: Питер, 2007 – 366 стр.