

Министерство образования и науки Астраханской области  
Государственное автономное образовательное учреждение  
Астраханской области высшего образования  
«Астраханский государственный архитектурно-строительный  
университет»  
(ГАОУ АО ВО «АГАСУ»)

---



### РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Наименование дисциплины Технологии программирования  
*(указывается наименование в соответствии с учебным планом)*

По направлению подготовки  
09.03.02 «ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ»  
*(указывается наименование направления подготовки в соответствии с ФГОС)*

Направленность (профиль)  
«Информационные системы и технологии в строительстве и архитектуре»  
*(указывается наименование профиля в соответствии с ОПОП)*

Кафедра Системы автоматизированного проектирования и моделирования

Квалификация выпускника бакалавр

Разработчик:

К.Т.И. Доцент  
(занимаемая должность,  
ученая степен, ученое звание)

[Подпись]  
(подпись)

Т.Н. Сапчиков  
(инициалы, фамилия)

Рабочая программа рассмотрена и утверждена на заседании кафедры «Системы автоматизированного проектирования и моделирования»

Протокол № 10 от 25.05 2019 г.

Заведующий кафедрой [Подпись] / Т.В. Хоменко  
(подпись)

Согласовано:

Председатель МКН «Информационные системы и технологии»  
направленность (профиль)  
«Информационные системы и технологии в строительстве и архитектуре»

[Подпись] Т.В. Хоменко  
(подпись) (инициалы, фамилия)

Начальник УМУ [Подпись] / И.В. Асеева  
(подпись) (инициалы, фамилия)

Специалист УМУ [Подпись] / Д.А. Фудина  
(подпись) (инициалы, фамилия)

Начальник УИТ [Подпись] / С.В. Туркина  
(подпись) (инициалы, фамилия)

Заведующий научной библиотекой [Подпись] / И.С. Кайдикешова  
(подпись) (инициалы, фамилия)

## Содержание

1. Цель освоения дисциплины.....	4
2. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы .....	4
3. Место дисциплины в структуре ОПОП бакалавриата .....	5
4. Объем дисциплины в зачетных единицах с указанием количества академических часов, выделенных на контактную работу обучающихся с преподавателем (по типам учебных занятий) и на самостоятельную работу обучающихся.....	5
5. Содержание дисциплины, структурированное по разделам с указанием отведенного на них количества академических часов и типов учебных занятий.....	6
5.1. Разделы дисциплины и трудоемкость по типам учебных занятий и работы обучающихся (в академических часах) .....	6
5.1.1. Очная форма обучения .....	6
5.1.2. Заочная форма обучения.....	6
5.2. Содержание дисциплины, структурированное по разделам.....	7
5.2.1. Содержание лекционных занятий.....	7
5.2.2. Содержание лабораторных занятий.....	8
5.2.3. Содержание практических занятий .....	8
5.2.4. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине .....	9
5.2.5. Темы контрольных работ .....	10
5.2.6. Темы курсовых работ.....	10
6. Методические указания для обучающихся по освоению дисциплины .....	10
7. Образовательные технологии .....	11
8. Учебно-методическое и информационное обеспечение дисциплины .....	12
8.1. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины .....	12
8.2. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения	13
8.3. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины .....	13
9. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине .....	14
10. Особенности организации обучения по дисциплине для инвалидов и лиц с ограниченными возможностями здоровья .....	14

## **1. Цель освоения дисциплины**

Целью освоения дисциплины «Технологии программирования» является формирование компетенций обучающихся в соответствии с требованиями Федерального государственного образовательного стандарта высшего образования по направлению подготовки 09.03.02 «Информационные системы и технологии».

## **2. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы**

В результате освоения дисциплины обучающийся должен овладеть следующими компетенциями:

ОПК-1 – Способен применять естественнонаучные и общеинженерные знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности;

ОПК-3 – Способен решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности;

ПК-2 – Способность выполнять интеграцию программных модулей и компонент.

В результате освоения дисциплин, формирующих компетенции ОПК-1, ОПК-3, ПК-2, обучающийся должен овладеть следующими результатами обучения:

знать:

– основы математики, физики, вычислительной техники и программирования (ОПК-1.1);

– принципы, методы и средства решения стандартных задач профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности (ОПК-3.1);

– методы и средства преобразования данных, языки, утилиты, среды программирования и сборки, интеграции программных модулей (компонент), методы и приемы формализации задач (ПК-2.1);

уметь:

– решать стандартные профессиональные задачи с применением естественнонаучных и общеинженерных знаний, методов математического анализа и моделирования (ОПК-1.2.);

– решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности (ОПК-3.2);

– писать программный код процедур интеграции программных модулей (ПК-2.2);

иметь навыки:

– теоретического и экспериментального исследования объектов профессиональной деятельности (ОПК-1.3);

– подготовки обзоров, аннотаций, составления рефератов, научных докладов, публикаций и библиографии по научно-исследовательской работе с учетом требований информационной безопасности (ОПК-3.3);

иметь практический опыт:  
 – выполнения процедур сборки программных модулей и компонент в программный продукт (ПК-2.3).

### 3. Место дисциплины в структуре ОПОП бакалавриата

Дисциплина Б1.О.12 «Технологии программирования» реализуется в рамках Блока 1 «Дисциплины (модули)» обязательной части.

Дисциплина базируется на знаниях, полученных в рамках изучения следующих дисциплин: «Информатика», «Алгоритмы и структуры данных», «Основы программирования».

### 4. Объем дисциплины в зачетных единицах с указанием количества академических часов, выделенных на контактную работу обучающихся с преподавателем (по типам учебных занятий) и на самостоятельную работу обучающихся

Форма обучения	Очная	Заочная
1	2	3
Трудоемкость в зачетных единицах:	3 семестр – 5 з.е.; всего – 5 з.е.	5 семестр – 5 з.е.; всего - 5 з.е.
Аудиторных (включая контактную работу обучающихся с преподавателем) часов (всего) по учебному плану:		
Лекции (Л)	3 семестр – 34 часа; всего - 34 часа	5 семестр – 8 часов; всего - 8 часов
Лабораторные занятия (ЛЗ)	3 семестр – 34 часа; всего - 34 часа	5 семестр – 12 часов; всего - 12 часов
Практические занятия (ПЗ)	учебным планом не предусмотрены	учебным планом не предусмотрены
Самостоятельная работа (СР)	3 семестр – 112 часов (в т.ч. КР – 36 часов); всего – 112 часов	5 семестр – 160 часов (в т.ч. КР – 36 часов); всего - 160 часов
Форма текущего контроля:		
Контрольная работа	учебным планом не предусмотрена	учебным планом не предусмотрена
Форма промежуточной аттестации:		
Зачет	учебным планом не предусмотрен	учебным планом не предусмотрен
Экзамен	семестр – 3	семестр – 5
Зачет с оценкой	учебным планом не предусмотрен	учебным планом не предусмотрен
Курсовая работа	семестр – 3	семестр – 5
Курсовой проект	учебным планом не предусмотрен	учебным планом не предусмотрен

**5. Содержание дисциплины, структурированное по разделам с указанием отведенного на них количества академических часов и типов учебных занятий**

**5.1. Разделы дисциплины и трудоемкость по типам учебных занятий и работы обучающихся (в академических часах)**

**5.1.1. Очная форма обучения**

№ п/п	Раздел дисциплины (по семестрам)	Всего часов на раздел	Семестр	Распределение трудоемкости раздела (в часах) по типам учебных занятий и работы обучающихся				Форма текущего контроля и промежуточной аттестации
				контактная			СР	
				Лекции	Лабор. занятия	Практ. занятия		
1	2	3	4	5	6	7	8	9
1	Раздел 1. Базовые понятия, методы и приемы объектно-ориентированного программирования	72	3	16	16	-	40	Курсовая работа
2	Раздел 2. Визуальные компоненты среды программирования C++	36		6	6	-	24	
3	Раздел 3. Основы работы с базами данных	72		12	12	-	48	
<b>Итого:</b>		<b>180</b>		<b>34</b>	<b>34</b>	<b>-</b>	<b>112</b>	

**5.1.2. Заочная форма обучения**

№ п/п	Раздел дисциплины (по семестрам)	Всего часов на раздел	Семестр	Распределение трудоемкости раздела (в часах) по типам учебных занятий и работы обучающихся				Форма текущего контроля и промежуточной аттестации
				контактная			СР	
				Лекции	Лабор. занятия	Практ. занятия		
1	2	3	4	5	6	7	8	9
1	Раздел 1. Базовые понятия, методы и приемы объектно-ориентированного программирования	72	5	4	4	-	64	Курсовая работа
2	Раздел 2. Визуальные компоненты среды программирования C++	36		2	4	-	30	
3	Раздел 3. Основы работы с базами данных	72		2	4	-	66	
<b>Итого:</b>		<b>180</b>		<b>8</b>	<b>12</b>	<b>-</b>	<b>160</b>	

## 5.2. Содержание дисциплины, структурированное по разделам

### 5.2.1. Содержание лекционных занятий

№	Наименование раздела дисциплины	Содержание
1.	Раздел 1. Базовые понятия, методы и приемы объектно-ориентированного программирования	<p>Основы вычислительной техники и особенности объектно-ориентированных языков программирования. Структура программы на языке C++. Принципы, методы и средства решения стандартных задач профессиональной деятельности на основе использования: операторов присваивания, ветвления, цикла и варианта; простые и составные типы данных. Правила оформления и работа с массивами данных. Стандартные функции и процедуры языка C++. Методы и средства преобразования сложных и структурированных типов данных. Строки и тексты. Модульная структура программ. Динамические структуры данных. Структуры, указатели и рекурсивные типы данных. Программирование линейных списков. Элементы объектно-ориентированного программирования. Классы. Наследование. Полиморфизм и динамические объекты. Подготовка обзоров, научных докладов и публикаций по научно-исследовательской работе с учетом требований информационной безопасности.</p>
2.	Раздел 2. Визуальные компоненты среды программирования C++	<p>Методы и средства представления, преобразования и интеграции программных визуальных компонентов. Форма – главный компонент приложения. Особенности модальных форм. Фактические и формальные параметры функций языка C++. Глобальные и локальные переменные подпрограмм. Программный код процедур интеграции программных модулей. Рекурсивные процедуры и функции. Однострочный и многострочный редакторы. Простой и комбинированный списки. Сложные элементы интерфейса. Графический режим программы. Средства для работы с файлами и каталогами. Процедуры сборки программных модулей и компонент в программный продукт.</p>
3.	Раздел 3. Основы работы с базами данных	<p>Методы и приемы формализации задач посредством построения баз данных и их приложений. Реляционные базы данных и средства работы с ними. Проектирование баз данных. Приведение к нормальным формам. Компоненты доступа к данным. Объекты поля. Визуальные компоненты для работы с данными. Механизмы навигационного доступа к данным. Основные возможности и средства языка SQL. Удаленные базы данных. Архитектура «клиент-сервер». Разработка, отладка и модификация программного продукта с учетом основных требований информационной безопасности.</p>

### 5.2.2. Содержание лабораторных занятий

№	Наименование раздела дисциплины	Содержание
1.	Раздел 1. Базовые понятия, методы и приемы объектно-ориентированного программирования	Лабораторная работа № 1 Методы и средства решения стандартных задач профессиональной деятельности на основе выделения классов и объектов в языке С++
		Лабораторная работа № 2 Стандартные задачи профессиональной деятельности на основе наследования и виртуальных функций
		Лабораторная работа № 3 Теоретическое и экспериментальное исследование объектов профессиональной деятельности с определением их иерархии и групп итераторов
		Лабораторная работа № 4 Обработка событий с учетом основных требований информационной безопасности
2.	Раздел 2. Визуальные компоненты среды программирования С++	Лабораторная работа № 5 Средства представления, преобразования и интеграции программных визуальных компонентов при перегрузке операций
		Лабораторная работа № 6 Методы и средства сборки и интеграции программных компонент при использовании шаблонов функций и классов
3.	Раздел 3. Основы работы с базами данных	Лабораторная работа № 7 Процедуры сборки программных модулей и компонент при использовании потоковых классов
		Лабораторная работа № 8 Программный код процедур программных модулей с подключением стандартной библиотеки шаблонов

### 5.2.3. Содержание практических занятий

Учебным планом не предусмотрены.



### 5.2.4. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине

#### Очная форма обучения

№	Наименование раздела дисциплины	Содержание	Учебно-методическое обеспечение
1	2	3	4
1.	Раздел 1. Базовые понятия, методы и приемы объектно-ориентированного программирования	Изучение теоретического материала по рекомендованной в рабочей программе литературе. Подготовка к выполнению и защите курсовой работы. Подготовка к экзамену.	[1], [3], [5], [8]
2.	Раздел 2. Визуальные компоненты среды программирования C++	Подготовка к выполнению лабораторных работ по написанию программ на языке C++. Подготовка к защите лабораторных работ. Подготовка к экзамену.	[2], [4], [7], [8]
3.	Раздел 3. Основы работы с базами данных	Изучение теоретического и практического материала по рекомендованной в рабочей программе литературе. Подготовка к выполнению лабораторных работ по отладке и модификации программ на C++. Подготовка к выполнению и защите курсовой работы. Подготовка к экзамену.	[1], [3], [4], [5], [6] [7]

#### Заочная форма обучения

№	Наименование раздела дисциплины	Содержание	Учебно-методическое обеспечение
1	2	3	4
1.	Раздел 1. Базовые понятия, методы и приемы объектно-ориентированного программирования	Изучение теоретического материала по рекомендованной в рабочей программе литературе. Подготовка к выполнению и защите курсовой работы. Подготовка к экзамену.	[1], [3], [5], [8]
2.	Раздел 2. Визуальные компоненты среды программирования C++	Подготовка к выполнению лабораторных работ по написанию программ на языке C++. Подготовка к защите лабораторных работ. Подготовка к экзамену.	[2], [4], [6], [8]
3.	Раздел 3. Основы работы с базами данных	Изучение теоретического и практического материала по рекомендованной в рабочей программе литературе. Подготовка к выполнению лабораторных работ по отладке и модификации программ на C++. Подготовка к выполнению и защите курсовой работы. Подготовка к экзамену.	[1], [3], [4], [5], [6] [8]

## 5.2.5. Темы контрольных работ

Учебным планом не предусмотрены.

## 5.2.6. Темы курсовых работ

1. Методы и средства решения стандартных задач профессиональной деятельности на основе построения графического редактора “Схемы алгоритмов”.
2. Методы и средства решения стандартных задач профессиональной деятельности на основе построения системы представления табличной информации в графическом виде.
3. Методы и средства решения стандартных задач профессиональной деятельности на основе построения специализированного текстового редактора.
4. Методы и средства решения стандартных задач профессиональной деятельности на основе построения системы расчета простейших электрических схем.
5. Методы и средства решения стандартных задач профессиональной деятельности на основе построения информационно-поисковой системы “Астроном”.
6. Методы и средства решения стандартных задач профессиональной деятельности на основе построения системы учета успеваемости студентов.
7. Методы и средства решения стандартных задач профессиональной деятельности на основе построения автоматизированной системы складского учета.

## 6. Методические указания для обучающихся по освоению дисциплины

Организация деятельности студента
<p><u>Лекция</u></p> <p>В ходе лекционных занятий необходимо вести конспектирование учебного материала, обращать внимание на категории, формулировки, раскрывающие содержание тех или иных явлений и процессов, научные выводы и практические рекомендации. Необходимо задавать преподавателю уточняющие вопросы с целью уяснения теоретических положений, разрешения спорных ситуаций. Целесообразно дорабатывать свой конспект лекции, делая в нем соответствующие записи из литературы, рекомендованной преподавателем и предусмотренной учебной программой.</p>
<p><u>Лабораторное занятие</u></p> <p>Работа в соответствии с методическими указания по выполнению лабораторных работ.</p>
<p><u>Самостоятельная работа</u></p> <p>Самостоятельная работа студента над усвоением учебного материала по учебной дисциплине может выполняться в помещениях для самостоятельной работы, а также в домашних условиях. Содержание самостоятельной работы студента определяется учебной программой дисциплины, методическими материалами, заданиями и указаниями преподавателя.</p> <p>Самостоятельная работа в аудиторное время может включать:</p> <ul style="list-style-type: none"><li>– конспектирование (составление тезисов) лекций;</li><li>– работу со справочной и методической литературой;</li><li>– участие в тестировании и др.</li></ul> <p>Самостоятельная работа во внеаудиторное время может состоять из:</p> <ul style="list-style-type: none"><li>– повторения лекционного материала;</li><li>– подготовки к лабораторным занятиям;</li><li>– выполнения курсовых работ, предусмотренных учебным планом;</li><li>– изучения учебной и научной литературы;</li><li>– выполнения заданий, выданных на лабораторных занятиях;</li><li>– подготовки к тестированию и т.д.;</li><li>– выделения наиболее сложных и проблемных вопросов по изучаемой теме, получения разъяснений и рекомендаций по данным вопросам от преподавателей кафедры на их еженедельных консультациях.</li></ul>

- проведения самоконтроля путем ответов на вопросы текущего контроля знаний, решения представленных в учебно-методических материалах кафедры задач и тестов.

#### Курсовая работа

Теоретическая часть курсовой работы выполняется по установленным темам с использованием практических материалов, полученных на лабораторных занятиях и при прохождении практики.

К каждой теме курсовой работы рекомендуется примерный перечень основных вопросов, список необходимой литературы. Необходимо изучить литературу, рекомендуемую для выполнения курсовой работы. Чтобы полнее раскрыть тему, следует использовать дополнительные источники и материалы. При написании курсовой работы необходимо ознакомиться с публикациями по теме, опубликованными в журналах.

Необходимо изложить собственные соображения по существу излагаемых вопросов, внести свои предложения. Общие положения должны быть подкреплены и пояснены конкретными примерами. Излагаемый материал при необходимости следует проиллюстрировать таблицами, схемами, диаграммами и т.д. Инструкция по выполнению требований к оформлению курсовой работы находится в методических материалах по дисциплине.

#### Подготовка к экзамену

Подготовка студентов к экзамену включает три стадии:

- самостоятельная работа в течение учебного семестра;
- непосредственная подготовка в дни, предшествующие экзамену;
- подготовка к ответу на вопросы, содержащиеся в билете.

## **7. Образовательные технологии**

Перечень образовательных технологий, используемых при изучении дисциплины «Технологии программирования».

### **Традиционные образовательные технологии**

Обучение дисциплине «Технологии программирования» проводится с использованием традиционных образовательных технологий, ориентирующихся на организацию образовательного процесса, предполагающую прямую трансляцию знаний от преподавателя к студенту (преимущественно на основе объяснительно-иллюстративных методов обучения). Учебная деятельность студента носит в таких условиях, как правило, репродуктивный характер. Формы учебных занятий по дисциплине «Технологии программирования» с использованием традиционных технологий:

Лекция – последовательное изложение материала в дисциплинарной логике, осуществляемое преимущественно вербальными средствами (монолог преподавателя).

Лабораторное занятие – организация учебной работы с реальными материальными и информационными объектами, экспериментальная работа с аналоговыми моделями реальных объектов.

### **Интерактивные технологии**

По дисциплине «Технологии программирования» лекционные занятия проводятся с использованием следующих интерактивных технологий:

Лекция-визуализация - представляет собой визуальную форму подачи лекционного материала средствами ТСО или аудиовидеотехники (видео-лекция). Чтение такой лекции сводится к развернутому или краткому комментированию просматриваемых визуальных материалов (в виде схем, таблиц, графов, графиков, моделей). Лекция-визуализация помогает студентам преобразовывать лекционный материал в визуальную форму, что способствует формированию у них профессионального мышления за счет систематизации и выделения наиболее значимых, существенных элементов.

Проблемная лекция – форма изложения материала, предполагающее постановку проблемных и дискуссионных вопросов, освещение различных научных подходов, авторские комментарии, связанные с различными моделями интерпретации изучаемого материала.

Лекция с разбором конкретных ситуаций – форма, при которой преподаватель на обсуждение ставит не вопросы, а конкретную ситуацию. Ситуация представляется устно или в очень короткой видеозаписи, диафильме, содержащих достаточную информацию для оценки характерного явления и обсуждения. Слушатели анализируют и обсуждают ее сообща, всей аудиторией. Основным содержанием занятия является лекционный материал, а потому преподаватель направляет тему дискуссии для получения достоверных выводов.

По дисциплине «Технологии программирования» лабораторные занятия проводятся с использованием следующих интерактивных технологий:

Работа в малых группах – это одна из самых популярных стратегий, так как она дает всем обучающимся возможность участвовать в работе, практиковать навыки сотрудничества, межличностного общения (в частности, умение активно слушать, вырабатывать общее мнение, разрешать возникающие разногласия).

Исследовательский проект – структура приближена к формату научного исследования (доказательство актуальности темы, определение научной проблемы, предмета и объекта исследования, целей и задач, методов, источников, выдвижение гипотезы, обобщение результатов, выводы, обозначение новых проблем).

Лабораторное занятие в форме практикума – организация учебной работы, направленная на решение комплексной учебно-познавательной задачи, требующей от студента применения как научно-теоретических знаний, так и практических навыков.

## **8. Учебно-методическое и информационное обеспечение дисциплины**

### **8.1. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины**

а) основная учебная литература:

1. Рик Гаско. Объектно-ориентированное программирование. Настольная книга программиста. Ред.: Комлев Н.Ю., изд.: Москва, Солон-пресс, 2018. – 298 с.

2. Гавриков М. М., Гринченков Д. В., Иванченко А. Н. Теоретические основы разработки и реализации языков программирования. Учеб. пособие. Изд.: Москва, Кнорус, 2016. -184 с.

3. Карпенков С. Х. Технические средства информационных технологий: учебное пособие - М., Берлин: Директ-Медиа, 2015. [http://biblioclub.ru/index.php?page=book\\_red&id=275367](http://biblioclub.ru/index.php?page=book_red&id=275367)

4. Грошев А.С. Информационные технологии: лабораторный практикум - М., Берлин: Директ-Медиа, 2015. [http://biblioclub.ru/index.php?page=book\\_red&id=434666](http://biblioclub.ru/index.php?page=book_red&id=434666)

б) дополнительная учебная литература:

5. Тарасов С. В. СУБД для программиста. Базы данных изнутри. Изд.: Москва, СОЛОН-Пресс, 2018. – 320 с.

6. Майстренко А. В., Майстренко Н. В. Информационные технологии в науке, образовании и инженерной практике: учебное пособие - Тамбов: Издательство ФГБОУ ВПО «ТГТУ», 2014. [http://biblioclub.ru/index.php?page=book\\_red&id=277993](http://biblioclub.ru/index.php?page=book_red&id=277993)

в) перечень учебно-методического обеспечения:

7. Садчиков, П.Н. Методические указания по выполнению лабораторных работ по дисциплине «Технологии программирования». АГАСУ. 2019. 29 с. <http://edu.aucu.ru>

8. Садчиков, П.Н. Методические указания к курсовой работе по дисциплине «Технологии программирования». АГАСУ. 2019. 47 с. <http://edu.aucu.ru>

г) перечень онлайн курсов:

9. Инженерия программного обеспечения  
[\\_https://www.intuit.ru/studies/higher\\_education/3406/info](https://www.intuit.ru/studies/higher_education/3406/info)

## **8.2. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения**

- 7-Zip
- Office 365 A1
- Adobe Acrobat Reader DC
- Google Chrome
- VLC media player
- Apache Open Office
- Office Pro Plus Russian OLPNL Academic Edition
- Kaspersky Endpoint Security
- Internet Explorer
- Visual Studio
- Microsoft SQL Server 2016 Express
- Microsoft Azure Dev Tools for Teaching
- Lazarus
- PascalABC.NET

## **8.3. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины**

1. Электронная информационно-образовательная среда Университета:

1.1. Образовательный портал (<http://edu.aucu.ru>, <http://moodle.aucu.ru>);

2. Электронно-библиотечные системы:

2.1. «Электронно-библиотечная система «Университетская библиотека» (<https://biblioclub.ru/>);

2.2. «Электронно-библиотечная система «IPRbooks» (<http://www.iprbookshop.ru/>)

3. Электронные базы данных:

3.1. Научная электронная библиотека (<http://www.elibrary.ru/>).

## 9. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине

№ п/п	Наименование специальных помещений и помещений для самостоятельной работы	Оснащенность специальных помещений и помещений для самостоятельной работы
1.	Учебная аудитория для проведения учебных занятий:  414056, г. Астрахань, ул. Татищева, 18, аудитория №207, №209, №211	аудитория №207  Комплект учебной мебели Компьютеры – 15 шт. Стационарный мультимедийный комплект Доступ к информационно – телекоммуникационной сети «Интернет»
		аудитория №209  Комплект учебной мебели Компьютеры – 15 шт. Стационарный мультимедийный комплект Доступ к информационно – телекоммуникационной сети «Интернет»
		аудитория №211  Комплект учебной мебели Компьютеры – 15 шт. Стационарный мультимедийный комплект Доступ к информационно – телекоммуникационной сети «Интернет»
2.	Помещение для самостоятельной работы:  414056, г. Астрахань, ул. Татищева, 18, аудитория №201  414056, г. Астрахань, ул. Татищева, 18б, аудитория №308	аудитория №201  Комплект учебной мебели Компьютеры – 4 шт. Доступ к информационно – телекоммуникационной сети «Интернет»
		аудитория №308  Комплект учебной мебели Компьютеры – 11 шт. Доступ к информационно – телекоммуникационной сети «Интернет»

## 10. Особенности организации обучения по дисциплине для инвалидов и лиц с ограниченными возможностями здоровья

Для обучающихся из числа инвалидов и лиц с ограниченными возможностями здоровья на основании письменного заявления дисциплина «Технологии программирования» реализуется с учетом особенностей психофизического развития, индивидуальных возможностей и состояния здоровья.

**Лист внесения дополнений и изменений  
в рабочую программу учебной дисциплины**

«Технологии программирования»

(наименование дисциплины)

на 20\_\_ - 20\_\_ учебный год

Рабочая программа пересмотрена на заседании кафедры  
«Системы автоматизированного проектирования и моделирования»,

протокол № \_\_\_\_ от \_\_\_\_\_ 20\_\_ г.

Зав. кафедрой

\_\_\_\_\_  
ученая степень, ученое звание

\_\_\_\_\_  
подпись

/\_\_\_\_\_  
И.О. Фамилия

В рабочую программу вносятся следующие изменения:

1. \_\_\_\_\_
2. \_\_\_\_\_
3. \_\_\_\_\_
4. \_\_\_\_\_
5. \_\_\_\_\_

Составители изменений и дополнений:

\_\_\_\_\_  
ученая степень, ученое звание

\_\_\_\_\_  
подпись

/\_\_\_\_\_  
И.О. Фамилия

\_\_\_\_\_  
ученая степень, ученое звание

\_\_\_\_\_  
подпись

/\_\_\_\_\_  
И.О. Фамилия

Председатель МКН «Информационные системы и технологии»

профиль «Информационные системы и технологии в строительстве и архитектуре»

\_\_\_\_\_  
ученая степень, ученое звание

\_\_\_\_\_  
подпись

/\_\_\_\_\_  
И.О. Фамилия

« \_\_\_\_ » \_\_\_\_\_ 20\_\_ г.

Министерство образования и науки Астраханской области  
Государственное автономное образовательное учреждение  
Астраханской области высшего образования  
«Астраханский государственный архитектурно-строительный  
университет»  
(ГАОУ АО ВО «АГАСУ»)

---



## ОЦЕНОЧНЫЕ И МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ

Наименование дисциплины Технологии программирования  
*(указывается наименование в соответствии с учебным планом)*

По направлению подготовки  
09.03.02 «ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ»  
*(указывается наименование направления подготовки в соответствии с ФГОС)*

Направленность (профиль)  
«Информационные системы и технологии в строительстве и архитектуре»  
*(указывается наименование профиля в соответствии с ОПОП)*

Кафедра Системы автоматизированного проектирования и моделирования

Квалификация выпускника бакалавр



Разработчик:

К.Т.Н. Дюбенг

(занимаемая должность,  
ученая степень, ученое звание)

[подпись]

(подпись)

Т.И. Садыков

(инициалы, фамилия)

Оценочные и методические материалы рассмотрены и утверждены на заседании кафедры  
«Системы автоматизированного проектирования и моделирования»

Протокол № 10 от 25.05 2019 г.

Заведующий кафедрой [подпись] / Т.В. Хоменко

(подпись)

Согласовано:

Председатель МКН «Информационные системы и технологии»  
направленность (профиль)  
«Информационные системы и технологии в строительстве и архитектуре»

[подпись] / Т.В. Хоменко

(подпись) (инициалы, фамилия)

Начальник УМУ

[подпись]

(подпись)

Н.В. Ахметжан

(инициалы, фамилия)

Специалист УМУ

[подпись]

(подпись)

Т.А. Душкина

(инициалы, фамилия)

## Содержание

1. Оценочные и методические материалы для проведения текущего контроля успеваемости и промежуточной аттестации обучающихся по дисциплине .....	4
1.1. Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы.....	4
1.2. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания.....	6
1.2.1. Перечень оценочных средств текущего контроля успеваемости.....	6
1.2.2. Описание показателей и критериев оценивания компетенций по дисциплине на различных этапах их формирования, описание шкал оценивания .....	6
1.2.3. Шкала оценивания .....	9
2. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков, характеризующих этапы формирования компетенций в процессе освоения образовательной программы.....	10
2.1. Экзамен .....	10
2.2. Курсовая работа.....	11
2.3. Защита лабораторной работы .....	11
2.4. Тест .....	12
Приложение 1 .....	14
Приложение 2.....	16

# 1. Оценочные и методические материалы для проведения текущего контроля успеваемости и промежуточной аттестации обучающихся по дисциплине

Оценочные и методические материалы являются неотъемлемой частью рабочей программы дисциплины (далее РПД) и представлены в виде отдельного документа.

## 1.1. Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы

Индекс и формулировка компетенции N	Индикаторы достижений компетенций, установленные ОПОП	Номер раздела дисциплины (в соответствии с п.5.1 РПД)			Формы контроля с конкретизацией задания
		1	2	3	
1	2	3	4	5	6
ОПК-1 – Способен применять естественнонаучные и общеинженерные знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности	Знать:				экзамен вопросы 1-19  курсовая работа темы 1-6  защита лабораторных работ (№1, №2)  тестирование вопросы 1-20
	основы математики, физики, вычислительной техники и программирования	X	X	X	
	Уметь:				
	решать стандартные профессиональные задачи с применением естественнонаучных и общеинженерных знаний, методов математического анализа и моделирования	X	X	X	
	Иметь навыки:				
	теоретического и экспериментального исследования объектов профессиональной деятельности	X	X	X	
ОПК-3 – Способен решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности	Знать:				экзамен вопросы 20-30  курсовая работа темы 7-13
	принципы, методы и средства решения стандартных задач профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности	X	X	X	

	Уметь:			защита лабораторных работ (№3, №4)  тестирование вопросы 21-42	
	решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности	X	X		X
Иметь навыки:					
подготовки обзоров, аннотаций, составления рефератов, научных докладов, публикаций и библиографии по научно-исследовательской работе с учетом требований информационной безопасности	X	X			X
ПК-2 – Способность выполнять интеграцию программных модулей и компонент	Знать:			экзамен вопросы 31-50  курсовая работа темы 14-20  защита лабораторных работ (№5, №6, №7, №8)  тестирование вопросы 43-62	
	методы и средства преобразования данных, языки, утилиты, среды программирования и сборки, интеграции программных модулей (компонент), методы и приемы формализации задач	X	X		X
	Уметь:				
	писать программный код процедур интеграции программных модулей	X	X		
	Иметь практический опыт:				
выполнения процедур сборки программных модулей и компонент в программный продукт	X	X	X		

## 1.2. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания

### 1.2.1. Перечень оценочных средств текущего контроля успеваемости

Наименование оценочного средства	Краткая характеристика оценочного средства	Представление оценочного средства в фонде
Защита лабораторной работы	Средство, позволяющее оценить умение и владение обучающегося излагать суть поставленной задачи, самостоятельно применять стандартные методы решения поставленной задачи с использованием имеющейся лабораторной базы, проводить анализ полученного результата работы. Рекомендуется для оценки умений и владений студентов	Темы лабораторных работ и требования к их защите
Тест	Система стандартизированных заданий, позволяющая автоматизировать процедуру измерения уровня знаний и умений обучающегося	Фонд тестовых заданий

### 1.2.2. Описание показателей и критериев оценивания компетенций по дисциплине на различных этапах их формирования, описание шкал оценивания

Компетенция, этапы освоения компетенции	Планируемые результаты обучения	Показатели и критерии оценивания результатов обучения			
		Ниже порогового уровня (не удовл.)	Пороговый уровень (удовл.)	Продвинутый уровень (хорошо)	Высокий уровень (отлично)
1	2	3	4	5	6
ОПК-1 – Способен применять естественнонаучные и общинженерные знания, методы математического анализа и моделирования,	Знает: основы математики, физики, вычислительной техники и программирования	Обучающийся не знает и не понимает основы математики, физики, вычислительной техники и программирования	Обучающийся не твердо знает и не вполне понимает основы математики, физики, вычислительной техники и программирования	Обучающийся знает и понимает основы математики, физики, вычислительной техники и программирования в типовых ситуациях для строительной сферы и архитектуры	Обучающийся знает и детально понимает основы математики, физики, вычислительной техники и программирования в ситуациях повышенной сложности для строительной сферы и архитектуры

теоретического и экспериментального исследования в профессиональной деятельности	Умеет: решать стандартные профессиональные задачи с применением естественнонаучных и общетехнических знаний, методов математического анализа и моделирования	Обучающийся не умеет решать стандартные профессиональные задачи с применением естественнонаучных и общетехнических знаний, методов математического анализа и моделирования	Обучающийся умеет решать стандартные профессиональные задачи с применением естественнонаучных и общетехнических знаний, методов математического анализа и моделирования	Обучающийся умеет решать стандартные профессиональные задачи с применением естественнонаучных и общетехнических знаний, методов математического анализа и моделирования в типовых ситуациях	Обучающийся умеет применять решать профессиональные задачи с применением естественнонаучных и общетехнических знаний, методов математического анализа и моделирования в ситуациях повышенной сложности
	Имеет навыки: теоретического и экспериментального исследования объектов профессиональной деятельности –	Обучающийся не имеет навыки теоретического и экспериментального исследования объектов профессиональной деятельности	Обучающийся имеет навыки теоретического и экспериментального исследования объектов профессиональной деятельности	Обучающийся имеет навыки теоретического и экспериментального исследования объектов профессиональной деятельности в типовых ситуациях	Обучающийся имеет навыки теоретического и экспериментального исследования объектов профессиональной деятельности в ситуациях повышенной сложности
ОПК-3 – Способен решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований	Знает: принципы, методы и средства решения стандартных задач профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности	Обучающийся не знает и не понимает принципы и методы решения стандартных задач профессиональной деятельности с применением информационно-коммуникационных технологий	Обучающийся не знает твердо и не вполне понимает принципы и методы решения задач профессиональной деятельности с применением информационно-коммуникационных технологий	Обучающийся знает и понимает структуру методов и средств решения стандартных задач профессиональной деятельности с применением информационно-коммуникационных технологий	Обучающийся знает и детально понимает структуру и назначение методов и средств решения задач профессиональной деятельности с применением информационно-коммуникационных технологий в ситуациях повышенной сложности

информационной безопасности	Умеет: решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности	Обучающийся не умеет решать стандартные задачи профессиональной деятельности с применением информационно-коммуникационных технологий	Обучающийся умеет решать стандартные задачи профессиональной деятельности с применением информационно-коммуникационных технологий по заранее известному алгоритму	Обучающийся умеет решать задачи профессиональной деятельности в типовых ситуациях с применением информационно-коммуникационных технологий	Обучающийся умеет решать задачи профессиональной деятельности с применением информационно-коммуникационных технологий в ситуациях повышенной сложности
	Имеет навыки: подготовки обзоров, аннотаций, составления рефератов, научных докладов, публикаций и библиографии по научно-исследовательской работе с учетом требований информационной безопасности	Обучающийся не имеет навыков подготовки обзоров и научных докладов по научно-исследовательской работе с учетом требований информационной безопасности	Обучающийся имеет навыки подготовки обзоров и научных докладов по научно-исследовательской работе с учетом требований информационной безопасности	Обучающийся имеет навыки подготовки обзоров и научных докладов по научно-исследовательской работе с учетом требований информационной безопасности	Обучающийся имеет навыки подготовки обзоров и научных докладов по научно-исследовательской работе с учетом требований информационной безопасности в отдельной предметной области
ПК-2 – Способность выполнять интеграцию программных модулей и компонент	Знает: методы и средства преобразования данных, языки, утилиты, среды программирования и сборки, интеграции программных модулей (компонент), методы и приемы формализации задач	Обучающийся не знает и не понимает назначение методов и средств преобразования данных при интеграции программных модулей, приемов их формализации	Обучающийся не знает твердо и не вполне понимает назначение методов и средств преобразования данных при интеграции программных модулей, приемов их формализации	Обучающийся знает и понимает назначение методов и средств преобразования данных при интеграции программных модулей, приемов их формализации в типовых ситуациях	Обучающийся знает и детально понимает назначение методов и средств преобразования данных при интеграции программных модулей, приемов их формализации в нестандартных ситуациях

	Умеет: писать программный код процедур интеграции программных модулей	Обучающийся не умеет писать программный код процедур интеграции программных модулей	Обучающийся умеет писать программный код процедур интеграции программных модулей по заранее известному алгоритму	Обучающийся умеет писать программный код процедур интеграции программных модулей под конкретную предметную область	Обучающийся умеет писать программный код процедур интеграции программных модулей в ситуациях повышенной сложности
	Имеет практический опыт: выполнения процедур сборки программных модулей и компонент в программный продукт	Обучающийся не имеет практического опыта выполнения процедур сборки программных модулей и компонент в программный продукт	Обучающийся имеет практический опыт выполнения процедур сборки программных модулей и компонент в программный продукт по заранее известному алгоритму	Обучающийся имеет практический опыт выполнения процедур сборки программных модулей и компонент в программный продукт под конкретную предметную область	Обучающийся имеет практический опыт выполнения процедур сборки программных модулей и компонент в программный продукт по широкому спектру областей знаний

### 1.2.3. Шкала оценивания

Уровень достижений	Отметка в 5-бальной шкале	Зачтено/ не зачтено
высокий	«5»(отлично)	зачтено
продвинутый	«4»(хорошо)	зачтено
пороговый	«3»(удовлетворительно)	зачтено
ниже порогового	«2»(неудовлетворительно)	не зачтено



**2. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков, характеризующих этапы формирования компетенций в процессе освоения образовательной программы**

**ТИПОВЫЕ ЗАДАНИЯ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ:**

**2.1. Экзамен**

- а) типовые вопросы к экзамену (см. приложение 1)
- б) критерии оценки

При оценке знаний на экзамене учитывается:

- 1. Уровень сформированности компетенций.
- 2. Уровень усвоения теоретических положений дисциплины, правильность формулировки основных понятий и закономерностей.
- 3. Уровень знания фактического материала в объеме программы.
- 4. Логика, структура и грамотность изложения вопроса.
- 5. Умение связать теорию с практикой.
- 6. Умение делать обобщения, выводы.

№ п/п	Оценка	Критерии оценки
1	Отлично	Ответы на поставленные вопросы излагаются логично, последовательно и не требуют дополнительных пояснений. Полно раскрываются причинно-следственные связи между явлениями и событиями. Делаются обоснованные выводы. Демонстрируются глубокие знания базовых нормативно-правовых актов. Соблюдаются нормы литературной речи.
2	Хорошо	Ответы на поставленные вопросы излагаются систематизировано и последовательно. Базовые нормативно-правовые акты используются, но в недостаточном объеме. Материал излагается уверенно. Раскрыты причинно-следственные связи между явлениями и событиями. Демонстрируется умение анализировать материал, однако не все выводы носят аргументированный и доказательный характер. Соблюдаются нормы литературной речи.
3	Удовлетворительно	Допускаются нарушения в последовательности изложения. Имеются упоминания об отдельных базовых нормативно-правовых актах. Неполно раскрываются причинно-следственные связи между явлениями и событиями. Демонстрируются поверхностные знания вопроса, с трудом решаются конкретные задачи. Имеются затруднения с выводами. Допускаются нарушения норм литературной речи.
4	Неудовлетворительно	Материал излагается непоследовательно, сбивчиво, не представляет определенной системы знаний по дисциплине. Не раскрываются причинно-следственные связи между явлениями и событиями. Не проводится анализ. Выводы отсутствуют. Ответы на дополнительные вопросы отсутствуют. Имеются заметные нарушения норм литературной речи.

## 2.2. Курсовая работа

а) типовые задания курсовой работы (см. приложение 1)

б) критерии оценки

При оценке знаний курсовой работы учитывается:

1. Уровень сформированности компетенций.
2. Уровень усвоения теоретических положений дисциплины, правильность формулировки основных понятий и закономерностей.
3. Уровень знания фактического материала в объеме программы.
4. Логика, структура и грамотность изложения вопроса.
5. Умение связать теорию с практикой.
6. Умение делать обобщения, выводы.

№ п/п	Оценка	Критерии оценки
1	2	3
1	Отлично	выставляется студенту, который: показывает всестороннее и глубокое освещение избранной темы в тесной взаимосвязи с практикой, а также умение работать с различными видами источников, систематизировать, классифицировать, обобщать материал, формулировать выводы, соответствующие поставленным целям.
2	Хорошо	выставляется студенту, который: обнаруживает глубокие знания по предмету и владеет навыками научного исследования, но при этом имеются незначительные замечания по содержанию работы, по процедуре защиты (студент не может дать аргументированно ответы на вопросы).
3	Удовлетворительно	выставляется студенту, который: неполно раскрывает разделы плана, посредственно владеет материалом, поверхностно отвечает на вопросы, в процессе защиты курсовой работы; отсутствуют аргументированные выводы, работа/проект носит реферативный характер.
4	Неудовлетворительно	выставляется студенту, если установлен акт самостоятельного выполнения работы, имеются принципиальные замечания по многим параметрам, содержание не соответствует теме, допущены грубые теоретические ошибки.

ТИПОВЫЕ ЗАДАНИЯ ДЛЯ ПРОВЕДЕНИЯ ТЕКУЩЕГО КОНТРОЛЯ:

## 2.3. Защита лабораторной работы

а) типовые задания лабораторных работ (см. приложение 2);

б) критерии оценки.

При оценке знаний на защите лабораторной работы учитывается:

1. Уровень сформированности компетенций.
2. Уровень усвоения теоретических положений дисциплины, правильность формулировки основных понятий и закономерностей.
3. Уровень знания фактического материала в объеме программы.
4. Логика, структура и грамотность изложения вопроса.
5. Умение связать теорию с практикой.
6. Умение делать обобщения, выводы.

№ п/п	Оценка	Критерии оценки
1	2	3
1	Отлично	Студент правильно называет метод исследования, правильно называет прибор, правильно демонстрирует методику исследования /измерения, правильно оценивает результат.
2	Хорошо	Студент правильно называет метод исследования, правильно называет прибор, допускает единичные ошибки в демонстрации методики исследования /измерения и оценке его результатов
3	Удовлетворительно	Студент неправильно называет метод исследования, но при этом дает правильное название прибора. Допускает множественные ошибки в демонстрации методики исследования /измерения и оценке его результатов
4	Неудовлетворительно	Студент неправильно называет метод исследования, дает неправильное название прибора. Не может продемонстрировать методику исследования /измерения, а также оценить результат

## 2.4. Тест

- а) типовые вопросы к проведению тестирования (см. приложение 2);  
б) критерии оценки.

При оценке знаний по результатам тестов учитывается:

1. Уровень сформированности компетенций.
2. Уровень усвоения теоретических положений дисциплины, правильность формулировки основных понятий и закономерностей.
3. Уровень знания фактического материала в объеме программы.
4. Логика, структура и грамотность изложения вопроса.
5. Умение связать теорию с практикой.
6. Умение делать обобщения, выводы.

№ п/п	Оценка	Критерии оценки
1	2	3
1	Отлично	если выполнены следующие условия: - даны правильные ответы не менее чем на 90% вопросов теста, исключая вопросы, на которые студент должен дать свободный ответ; на все вопросы, предполагающие свободный ответ, студент дал правильный и полный ответ.
2	Хорошо	если выполнены следующие условия: - даны правильные ответы не менее чем на 75% вопросов теста, исключая вопросы, на которые студент должен дать свободный ответ; - на все вопросы, предполагающие свободный ответ, студент дал правильный ответ, но допустил незначительные ошибки и не показал необходимой полноты.
3	Удовлетворительно	если выполнены следующие условия: - даны правильные ответы не менее чем на 50% вопросов теста, исключая вопросы, на которые студент должен дать свободный ответ; - на все вопросы, предполагающие свободный ответ, студент дал непротиворечивый ответ, или при ответе допустил значительные неточности и не показал полноты.

4	Неудовлетворительно	если студентом не выполнены условия, предполагающие оценку «Удовлетворительно».
5	Зачтено	Выставляется при соответствии параметрам экзаменационной шкалы на уровнях «отлично», «хорошо», «удовлетворительно».
6	Не зачтено	Выставляется при соответствии параметрам экзаменационной шкалы на уровне «неудовлетворительно».

### 3. Перечень и характеристики процедуры оценивания знаний, умений, навыков, характеризующих этапы формирования компетенций

Процедура проведения текущего контроля успеваемости и промежуточной аттестации обучающихся по дисциплине регламентируется локальным нормативным актом.

#### Перечень и характеристика процедур текущего контроля и промежуточной аттестации по дисциплине

№	Наименование оценочного средства	Периодичность и способ проведения процедуры оценивания	Виды выставляемых оценок	Форма учета
1.	Экзамен	Раз в семестр, по окончании изучения дисциплины	По пятибалльной шкале	Ведомость, зачетная книжка, портфолио
2.	Курсовая работа	Раз в семестр, по окончании изучения дисциплины	По пятибалльной шкале	Ведомость, зачетная книжка, портфолио
3.	Защита лабораторной работы	Систематически на занятиях	По пятибалльной шкале	Лабораторная тетрадь, журнал успеваемости преподавателя
4.	Тест	Систематически на занятиях	По пятибалльной шкале или зачтено / не зачтено	Журнал успеваемости преподавателя

Экзамен  
Типовые вопросы и задания

## ОПК-1

1. Основы вычислительной техники и языка программирования C++: локализация и кириллица в консоли.
2. Основы языка программирования C++: структура программы программирования.
3. Основы языка программирования C++: переменные.
4. Основы языка программирования C++: типы данных.
5. Основы программирования: статическая типизация и преобразования типов.
6. Основы математического анализа и программирования: константы, арифметические операции, условные выражения.
7. Основы математического анализа и программирования: побитовые операции. Операции присваивания.
8. Основы вычислительной техники: ввод и вывод в консоли, пространства имен и using.
9. Методы моделирования: оператор ветвления.
10. Методы моделирования: циклы.
11. Методы моделирования ссылок на объекты теоретического и экспериментального исследования в профессиональной деятельности
12. Методы моделирования, оформления и обработки массивов.
13. Основы программирования строчных редакторов.
14. Определение и объявление функций
15. Методы моделирования и описания параметров функций.
16. Передача аргументов по значению и по ссылке
17. Основы программирования константных параметров.
18. Основы программирования оператора return и возвращение результата.
19. Методы моделирования и оформления рекурсивных функций.

## ОПК-3

20. Средства решения стандартных задач профессиональной деятельности посредством выявления области видимости объектов.
21. Разделение программы на файлы с учетом основных требований информационной безопасности.
22. Построение внешних объектов с учетом основных требований информационной безопасности.
23. Принципы, методы и средства решения стандартных задач профессиональной деятельности на основе операций с указателями в константах
24. Принципы, методы и средства решения задач профессиональной деятельности на основе операций с указателями в константах
25. Константы и указатели
26. Принципы, методы и средства решения стандартных задач профессиональной деятельности на основе операций с указателями в константах
27. Принципы, методы и средства решения стандартных задач профессиональной деятельности на основе операций с указателями в массивах

28. Средства решения задач профессиональной деятельности на основе операций с указателями параметрах функции
29. Решение стандартных задач с применением информационно- коммуникационных технологий со ссылками на динамические объекты.
30. Стандартные задачи с применением динамических массивов.

## ПК-2

31. Определение классов при интеграции программных модулей и компонент.
32. Программный код процедур интеграции программных модулей средствами конструктора и инициализации объектов.
33. Методы и средства преобразования данных при объявлении и определении функций класса.
34. Методы и средства преобразования при управлении доступом к данным.
35. Программный код процедур интеграции программных модулей через дружественные функции и классы.
36. Структуры при интеграции программных модулей и компонент.
37. Методы и приемы формализации задач перечислениями.
38. Наследование при интеграции программных модулей и компонент.
39. Виртуальные функции и их переопределение.
40. Интеграция программных модулей средствами Деструктора.
41. Операторы преобразования типов
42. Обработка исключений при интеграции программных модулей и компонент.
43. Типы исключений при интеграции программных модулей и компонент.
44. Типы последовательных контейнеров при выполнении процедур сборки программных модулей и компонент.
45. Базовые типы для работы с потоками при интеграции программных модулей и компонент.
46. Программный код процедур интеграции программных модулей при работе с файловыми структурами.
47. Чтение и запись текстовых файлов при интеграции программных модулей и компонент.
48. Переопределение операторов ввода и вывода при интеграции программных модулей и компонент.
49. Программный код процедур интеграции программных модулей с определением шаблонов классов.
50. Программный код процедур интеграции функций.

## Курсовая работа Примерные темы

Тематика курсовых проектов по дисциплинам определяется преподавателем. При этом выбор основывается как на государственном стандарте, так и на направлениях научно-исследовательской и учебно-методической работы, актуальных направлениях работы других организаций, деятельность которых связана с разработкой, информационного и программного обеспечения ЭВМ. Студенту предоставляется право выбора одной из предложенных тем или предложения своей темы с обоснованием целесообразности ее разработки.

## ОПК-1

1. Основы программирования при моделировании электронных часов с будильником.
2. Основы программирования при реализации модели часов со стрелками.
3. Основы программирования анимации перемещений числовых объектов по форме.
4. Основы программирования анимации перемещений графических объектов по форме игры "Парные картинки".
5. Основы программирования при реализации графических объектов в игре "Сапер".
6. Основы программирования при реализации модели оценки способности игрока (испытуемого) запоминать числа.

## ОПК-3

7. Методы и средства решения стандартных задач профессиональной деятельности на основе построения графического редактора "Схемы алгоритмов".
8. Методы и средства решения стандартных задач профессиональной деятельности на основе построения системы представления табличной информации в графическом виде.
9. Методы и средства решения стандартных задач профессиональной деятельности на основе построения специализированного текстового редактора.
10. Методы и средства решения стандартных задач профессиональной деятельности на основе построения системы расчета простейших электрических схем.
11. Методы и средства решения стандартных задач профессиональной деятельности на основе построения информационно-поисковой системы "Астроном".
12. Методы и средства решения стандартных задач профессиональной деятельности на основе построения системы учета успеваемости студентов.
13. Методы и средства решения стандартных задач профессиональной деятельности на основе построения автоматизированной системы складского учета.

## ПК-2

14. Методы и средства интеграции программных модулей при разработке инженерного калькулятора.
15. Методы и средства интеграции программных модулей по изображению графиков математических функций.
16. Методы и средства интеграции программных модулей по выводу диаграмм табличных значений.
17. Методы и средства интеграции программных модулей медиа-плеера по проигрыванию аудио- и видеофайлов.
18. Методы и средства интеграции программных модулей в игре "Пятнашки".
19. Методы и средства интеграции программных модулей по проекту "Записная книжка".
20. Методы и средства интеграции программных модулей по проекту "Ежедневник".

Защита лабораторных работ  
Типовые задания

ОПК-1

Лабораторная работа № 1  
МЕТОДЫ И СРЕДСТВА РЕШЕНИЯ СТАНДАРТНЫХ ЗАДАЧ ПРОФЕССИОНАЛЬНОЙ  
ДЕЯТЕЛЬНОСТИ НА ОСНОВЕ ВЫДЕЛЕНИЯ КЛАССОВ И ОБЪЕКТОВ В ЯЗЫКЕ C++

Цель. Получить практические навыки реализации классов на C++.

Основное содержание работы.

Написать программу, в которой создаются и разрушаются объекты, определенного пользователем класса. Выполнить исследование вызовов конструкторов и деструкторов.

Порядок выполнения работы.

1. Определить пользовательский класс в соответствии с вариантом задания (смотри приложение).
2. Определить в классе следующие конструкторы: без параметров, с параметрами, копирования.
3. Определить в классе деструктор.
4. Определить в классе компоненты-функции для просмотра и установки полей данных.
5. Определить указатель на компоненту-функцию.
6. Определить указатель на экземпляр класса.
7. Написать демонстрационную программу, в которой создаются и разрушаются объекты пользовательского класса и каждый вызов конструктора и деструктора сопровождается выдачей соответствующего сообщения (какой объект какой конструктор или деструктор вызвал).
8. Показать в программе использование указателя на объект и указателя на компоненту-функцию.

Содержание отчета.

1. Титульный лист: название дисциплины; номер и наименование работы; фамилия, имя, отчество студента; дата выполнения.
2. Постановка задачи. Следует дать конкретную постановку, т.е. указать, какой класс должен быть реализован, какие должны быть в нем конструкторы, компоненты-функции и т.д.
3. Определение пользовательского класса с комментариями.
4. Реализация конструкторов и деструктора.
5. Фрагмент программы, показывающий использование указателя на объект и указателя на функцию с объяснением.
6. Листинг основной программы, в котором должно быть указано, в каком месте и какой конструктор или деструктор вызываются.



Лабораторная работа № 2  
СТАНДАРТНЫЕ ЗАДАЧИ ПРОФЕССИОНАЛЬНОЙ ДЕЯТЕЛЬНОСТИ НА ОСНОВЕ  
НАСЛЕДОВАНИЯ И ВИРТУАЛЬНЫХ ФУНКЦИЙ

Цель. Получить практические навыки создания иерархии классов и использования статических компонентов класса.

Основное содержание работы.

Написать программу, в которой создается иерархия классов. Включить полиморфные объекты в связанный список, используя статические компоненты класса. Показать использование виртуальных функций.

Порядок выполнения работы.

1. Определить иерархию классов (в соответствии с вариантом).
2. Определить в классе статическую компоненту - указатель на начало связанного списка объектов и статическую функцию для просмотра списка.
3. Реализовать классы.
4. Написать демонстрационную программу, в которой создаются объекты различных классов и помещаются в список, после чего список просматривается.
5. Сделать соответствующие методы не виртуальными и посмотреть, что будет.
6. Реализовать вариант, когда объект добавляется в список при создании, т.е. в конструкторе (смотри пункт 6 следующего раздела).

Содержание отчета.

1. Титульный лист: название дисциплины; номер и наименование работы; фамилия, имя, отчество студента; дата выполнения.
2. Постановка задачи. Следует дать конкретную постановку, т.е. указать, какие классы должны быть реализованы, какие должны быть в них конструкторы, компоненты-функции и т.д.
3. Иерархия классов в виде графа.
4. Определение пользовательских классов с комментариями.
5. Реализация конструкторов с параметрами и деструктора.
6. Реализация методов для добавления объектов в список.
7. Реализация методов для просмотра списка.
8. Листинг демонстрационной программы.

Варианты заданий.

Перечень классов:

- 1) студент, преподаватель, персона, завкафедрой;
- 2) служащий, персона, рабочий, инженер;
- 3) рабочий, кадры, инженер, администрация;
- 4) деталь, механизм, изделие, узел;
- 5) организация, страховая компания, судостроительная компания, завод;
- 6) журнал, книга, печатное издание, учебник;
- 7) тест, экзамен, выпускной экзамен, испытание;
- 8) место, область, город, мегаполис;
- 9) игрушка, продукт, товар, молочный продукт;
- 10) квитанция, накладная, документ, чек;
- 11) автомобиль, поезд, транспортное средство, экспресс;
- 12) двигатель, двигатель внутреннего сгорания, дизель, турбореактивный двигатель;
- 13) республика, монархия, королевство, государство;
- 14) млекопитающие, парнокопытные, птицы, животное;
- 15) корабль, пароход, парусник, корвет.

Лабораторная работа № 3  
ТЕОРЕТИЧЕСКОЕ И ЭКСПЕРИМЕНТАЛЬНОЕ ИССЛЕДОВАНИЕ ОБЪЕКТОВ  
ПРОФЕССИОНАЛЬНОЙ ДЕЯТЕЛЬНОСТИ С ОПРЕДЕЛЕНИЕМ ИХ ИЕРАРХИИ  
И ГРУПП ИТЕРАТОРОВ

Цель. Получить практические навыки создания объектов-групп и использования методов-итераторов.

Порядок выполнения работы.

1. Дополнить иерархию классов лабораторной работы № 2 классами “группа”.

Например, для предметной области ФАКУЛЬТЕТ можно предложить классы “факультет”, “студенческая группа”, “кафедра”. Рекомендуются создать абстрактный класс – “подразделение”, который будет предком всех групп и абстрактный класс TObject, находящийся во главе всей иерархии.

2. Написать для класса-группы метод-итератор.

3. Написать процедуру или функцию, которая выполняется для всех объектов, входящих в группу (смотри примеры в приложении).

4. Написать демонстрационную программу, в которой создаются, показываются и разрушаются объекты-группы, а также демонстрируется использование итератора.

Содержание отчета.

1. Титульный лист.

2. Постановка задачи.

3. Иерархия классов.

4. Иерархия объектов.

5. Определение классов (добавленных или измененных по сравнению с лабораторной работой № 2).

6. Реализация для одного не абстрактного класса-группы всех методов.

7. Реализация итератора.

8. Реализация передаваемой итератору функции.

9. Листинг демонстрационной программы.

Варианты запросов.

1. Имена всех лиц мужского (женского) пола.

2. Имена студентов указанного курса.

3. Имена и должность преподавателей указанной кафедры.

4. Имена служащих со стажем не менее заданного.

5. Имена служащих заданной профессии.

6. Имена рабочих заданного цеха.

7. Имена рабочих заданной профессии.

8. Имена студентов, сдавших все (заданный) экзамены на отлично (хорошо и отлично).

9. Имена студентов, не сдавших все (хотя бы один) экзамен.

10. Имена всех монархов на заданном континенте.

11. Наименование всех деталей (узлов), входящих в заданный узел (механизм).

12. Наименование всех книг в библиотеке (магазине), вышедших не ранее указанного года.

13. Названия всех городов заданной области.

14. Наименование всех товаров в заданном отделе магазина.

15. Количество мужчин (женщин).

16. Количество студентов на указанном курсе.

17. Количество служащих со стажем не менее заданного.

18. Количество рабочих заданной профессии.

19. Количество инженеров в заданном подразделении.
20. Количество товара заданного наименования.
21. Количество студентов, сдавших все экзамены на отлично.
22. Количество студентов, не сдавших хотя бы один экзамен.
23. Количество деталей (узлов), входящих в заданный узел (механизм).
24. Количество указанного транспортного средства в автопарке (на автостоянке).
25. Количество пассажиров во всех вагонах экспресса.
26. Суммарная стоимость товара заданного наименования.
27. Средний балл за сессию заданного студента.
28. Средний балл по предмету для всех студентов.
29. Суммарное количество учебников в библиотеке (магазине).
30. Суммарное количество жителей всех городов в области.
31. Суммарная стоимость продукции заданного наименования по всем накладным.
32. Средняя мощность всех (заданного типа) транспортных средств в организации.
33. Средняя мощность всех дизелей, обслуживаемых заданной фирмой.
34. Средний вес животных заданного вида в зоопарке.
35. Среднее водоизмещение всех парусников на верфи (в порту).

Лабораторная работа № 4  
**ОБРАБОТКА СОБЫТИЙ С УЧЕТОМ ОСНОВНЫХ ТРЕБОВАНИЙ  
ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ**

Цель. Получить практические навыки разработки объектно-ориентированной программы, управляемой событиями.

Основное содержание работы.

Написать интерактивную программу, выполняющую команды, вводимые пользователем с клавиатуры.

Порядок выполнения работы.

1. Разобрать пример, представленный в приложении. Ответить на следующие вопросы:
  - а) какова здесь иерархия классов?
  - б) какова здесь иерархия объектов?
  - в) как КАЛЬКУЛЯТОРУ передаются аргументы операции? где они хранятся? Каким образом получают к ним доступ устройства СЛОЖЕНИЯ, ВЫЧИТАНИЯ и т.д. ?
  - г) как обрабатываются события группой?
  - д) каковы все маршруты события TEvent?
  - е) как выполняются HandleEvent всех классов?
2. Выбрать группу объектов, которые будут обрабатывать события (это не могут быть объекты, приведенные в приложении).
3. Для выбранной группы объектов определить перечень операций, которые должны выполняться по командам пользователя.
4. Определить вид командной строки <код\_операции><параметры>. Решить вопросы: как кодируются операции? какие передаются параметры?
5. Определить иерархию объектов. Если необходимо, добавить новые объекты (группы объектов).
6. Определить иерархию классов. Если необходимо, добавить новые классы.
7. Определить, какой объект в программе играет роль приложения. В случае необходимости добавить в иерархию классов класс TApp. Решить, в каком классе будет метод Execute, организующий главный цикл обработки событий.
8. Определить и реализовать необходимые для обработки событий методы.
9. Написать основную функцию (main).

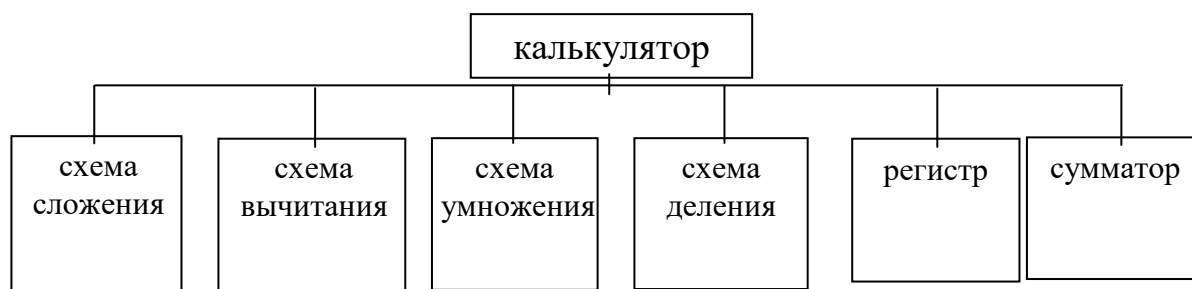
### Содержание отчета.

1. Титульный лист.
2. Постановка задачи.
3. Схема иерархии классов.
4. Схема иерархии объектов.
5. Описание маршрута, который проходит событие TEvent от формирования до очистки.
6. Определения классов.
7. Реализация методов обработки событий GetEvent, Execute, EndExec, Valid.
8. Реализация всех методов (для всех классов) HandleEvent.
9. Листинг функции main().

### Приложение.

Объект КАЛЬКУЛЯТОР выполняет сложение, вычитание, умножение, деление вещественных чисел.

### Иерархия объектов.



ПК-2

## Лабораторная работа № 5 СРЕДСТВА ПРЕДСТАВЛЕНИЯ, ПРЕОБРАЗОВАНИЯ И ИНТЕГРАЦИИ ПРОГРАММНЫХ ВИЗУАЛЬНЫХ КОМПОНЕНТОВ ПРИ ПЕРЕГРУЗКЕ ОПЕРАЦИЙ

Цель. Получить практические навыки работы в среде VC++5.02 и создания EasyWin-программы. Получить практические навыки создания абстрактных типов данных и перегрузки операций в языке C++.

### Основное содержание работы.

Определить и реализовать класс – абстрактный тип данных. Определить и реализовать операции над данными этого класса. Написать и выполнить Easy Win-программу полного тестирования этого класса.

### Порядок выполнения работы.

1. Выбрать класс АД в соответствии с вариантом.
2. Определить и реализовать в классе конструкторы, деструктор, функции Input (ввод с клавиатуры) и Print (вывод на экран), перегрузить операцию присваивания.
3. Написать программу тестирования класса и выполнить тестирование.
4. Дополнить определение класса заданными перегруженными операциями (в соответствии с вариантом).
5. Реализовать эти операции. Выполнить тестирование.

### Содержание отчета.

1. Титульный лист.
2. Конкретное задание с указанием номера варианта, реализуемого класса и операций.
3. Определение класса.
4. Обоснование включения в класс нескольких конструкторов, деструктора и операции присваивания.

5. Объяснить выбранное представление памяти для объектов реализуемого класса.
6. Реализация перегруженных операций с обоснованием выбранного способа (функция – член класса, внешняя функция, внешняя дружественная функция).
7. Тестовые данные и результаты тестирования.

#### Вопросы для самоконтроля.

1. Что такое абстрактный тип данных?
2. Приведите примеры абстрактных типов данных.
3. Каковы синтаксис/семантика “операции-функции”?
4. Как можно вызвать операцию-функцию?
5. Нужно ли перегружать операцию присваивания относительно определенного пользователем типа данных, например класса? Почему?
6. Можно ли изменить приоритет перегруженной операции?
7. Можно ли изменить количество операндов перегруженной операции?
8. Можно ли изменить ассоциативность перегруженной операции?
9. Можно ли, используя дружественную функцию, перегрузить оператор присваивания?
10. Все ли операторы языка C++ могут быть перегружены?
11. Какими двумя разными способами определяются перегруженные операции?
12. Все ли операции можно перегрузить с помощью глобальной дружественной функции?
13. В каких случаях операцию можно перегрузить только глобальной функцией?
14. В каких случаях глобальная операция-функция должна быть дружественной?
15. Обязателен ли в функции operator параметр типа “класс” или “ссылка на класс”?
16. Наследуются ли перегруженные операции?
17. Можно ли повторно перегрузить в производном классе операцию, перегруженную в базовом классе?
18. В чем отличие синтаксиса операции-функции унарной и бинарной операции?
19. Приведите примеры перегрузки операций для стандартных типов.
20. Перегрузите операцию “+” для класса “комплексное число”.
21. Перегрузите операции “<”, “>”, “==” для класса “строка символов”.

#### Варианты заданий

1. АТД – множество с элементами типа char. Дополнительно перегрузить следующие операции:
  - + – добавить элемент в множество(типа char + set);
  - + – объединение множеств;
  - = – проверка множеств на равенство.
2. АТД – множество с элементами типа char. Дополнительно перегрузить следующие операции:
  - – удалить элемент из множества (типа set-char);
  - \* – пересечение множеств;
  - < – сравнение множеств.
3. АТД – множество с элементами типа char. Дополнительно перегрузить следующие операции:
  - – удалить элемент из множества (типа set-char);
  - > – проверка на подмножество;
  - != – проверка множеств на неравенство.
4. АТД – множество с элементами типа char. Дополнительно перегрузить следующие операции:
  - + – добавить элемент в множество (типа set+char);
  - \* – пересечение множеств;
  - int()– мощность множества.

5. АД – множество с элементами типа char. Дополнительно перегрузить следующие операции:

- () – конструктор множества (в стиле конструктора Паскаля);
- + – объединение множеств;
- <= – сравнение множеств .

6. АД – множество с элементами типа char. Дополнительно перегрузить следующие операции:

- > – проверка на принадлежность(char in set Паскаля);
- \* – пересечение множеств;
- < – проверка на подмножество.

7. АД – однонаправленный список с элементами типа char. Дополнительно перегрузить следующие операции:

- + – объединить списки (list+list);
- – удалить элемент из начала (типа --list);
- = – проверка на равенство.

8. АД – однонаправленный список с элементами типа char. Дополнительно перегрузить следующие операции:

- + – добавить элемент в начало(char+list);
- – удалить элемент из начала(типа –list);
- = – проверка на равенство.

9. АД – однонаправленный список с элементами типа char. Дополнительно перегрузить следующие операции:

- + – добавить элемент в конец (list+char);
- – удалить элемент из конца (типа list--);
- != – проверка на неравенство.

10. АД – однонаправленный список с элементами типа char. Дополнительно перегрузить следующие операции:

- [] – доступ к элементу в заданной позиции, например:  
int i; char c;  
list L;  
c=L[i];
- + – объединить два списка;
- = – проверка на равенство.

11. АД – однонаправленный список с элементами типа char. Дополнительно перегрузить следующие операции:

- [] – доступ к элементу в заданной позиции, например:  
int i; char c;  
list L;  
c=L[i];
- + – объединить два списка;
- != – проверка на неравенство.

12. АД – однонаправленный список с элементами типа char. Дополнительно перегрузить следующие операции:

- () – удалить элемент в заданной позиции, например :  
int i;

list L;  
L[i];  
( ) – добавить элемент в заданную позицию, например :  
int i; char c;  
list L;  
L[c,i];  
!= – проверка на неравенство.

13. АДТ – стек. Дополнительно перегрузить следующие операции:

+ – добавить элемент в стек;  
– – извлечь элемент из стека;  
bool() – проверка, пустой ли стек.

14. АДТ – очередь. Дополнительно перегрузить следующие операции:

+ – добавить элемент;  
– – извлечь элемент;  
bool() – проверка, пустая ли очередь.

15. АДТ – одномерный массив (вектор) вещественных чисел. Дополнительно перегрузить следующие операции:

+ – сложение векторов ( $a[i]+b[i]$  для всех  $i$ );  
[] – доступ по индексу;  
+ – добавить число к вектору (double+vector).

16. АДТ – одномерный массив (вектор) вещественных чисел. Дополнительно перегрузить следующие операции:

- – вычитание векторов ( $a[i]-b[i]$  для всех  $i$ );  
[] – доступ по индексу;  
- – вычесть из вектора число (vector-double).

17. АДТ – одномерный массив (вектор) вещественных чисел. Дополнительно перегрузить следующие операции:

\* – умножение векторов ( $a[i]*b[i]$  для всех  $i$ );  
[] – доступ по индексу;  
\* – умножить вектор на число (vector\*double).

18. АДТ – одномерный массив (вектор) вещественных чисел. Дополнительно перегрузить следующие операции:

int() – размер вектора;  
( ) – установить новый размер;  
- – вычесть из вектора число (vector-double);  
[] – доступ по индексу;

19. АДТ – одномерный массив (вектор) вещественных чисел. Дополнительно перегрузить следующие операции:

= – присвоить всем элементам вектора значение (vector=double);  
[] – доступ по индексу;  
== – проверка на равенство;  
!= – проверка на неравенство;

20. АДТ – двумерный массив (матрица) вещественных чисел. Дополнительно перегрузить следующие операции:

- () – доступ по индексу;
- \* – умножение матриц;
- \* – умножение матрицы на число;
- \* – умножение числа на матрицу.

21. АДД – двумерный массив (матрица) вещественных чисел. Дополнительно перегрузить следующие операции:

- () – доступ по индексу;
- – разность матриц;
- – вычесть из матрицы число;
- = – проверка матриц на равенство.

22. АДД – двумерный массив (матрица) вещественных чисел. Дополнительно перегрузить следующие операции:

- () – доступ по индексу;
- = – присвоить всем элементам матрицы значение (matr=double);
- + – сложение матриц;
- + – сложить матрицу с числом (matr+double).

23. АДД – двумерный массив (матрица) вещественных чисел. Дополнительно перегрузить следующие операции:

- () – доступ по индексу;
- = – проверка матриц на равенство;
- ++ – транспонировать матрицу.

## Лабораторная работа № 6 МЕТОДЫ И СРЕДСТВА СБОРКИ И ИНТЕГРАЦИИ ПРОГРАММНЫХ КОМПОНЕНТ ПРИ ИСПОЛЬЗОВАНИИ ШАБЛОНОВ ФУНКЦИЙ И КЛАССОВ

Цель. Получить практические навыки создания шаблонов и использования их в программах C++.

### Основное содержание работы.

Создать шаблон заданного класса и использовать его для данных различных типов.

### Порядок выполнения работы.

1. Создать шаблон заданного класса. Определить конструкторы, деструктор, перегруженную операцию присваивания (“=”) и операции, заданные в варианте задания.
2. Написать программу тестирования, в которой проверяется использование шаблона для стандартных типов данных.
3. Выполнить тестирование.
4. Определить пользовательский класс, который будет использоваться в качестве параметра шаблона. Определить в классе необходимые функции и перегруженные операции.
5. Написать программу тестирования, в которой проверяется использование шаблона для пользовательского типа.
6. Выполнить тестирование.

### Содержание отчета.

1. Титульный лист: название дисциплины, номер и наименование работы, фамилия, имя, отчество студента, дата выполнения.
2. Постановка задачи.



Следует дать конкретную постановку, т.е. указать шаблон какого класса должен быть создан, какие должны быть в нем конструкторы, компоненты-функции, перегруженные операции и т.д.

То же самое следует указать для пользовательского класса.

3. Определение шаблона класса с комментариями.

4. Определение пользовательского класса с комментариями.

5. Реализация конструкторов, деструктора, операции присваивания и операций, которые заданы в варианте задания.

6. То же самое для пользовательского класса.

7. Результаты тестирования. Следует указать для каких типов и какие операции проверены и какие выявлены ошибки (или не выявлены)

#### Вопросы для самоконтроля.

1. В чем смысл использования шаблонов?

2. Каковы синтаксис/семантика шаблонов функций?

3. Каковы синтаксис/семантика шаблонов классов?

4. Напишите параметризованную функцию сортировки массива методом обмена.

5. Определите шаблон класса “вектор” – одномерный массив.

6. Что такое параметры шаблона функции?

7. Перечислите основные свойства параметров шаблона функции.

8. Как записывать параметр шаблона?

9. Можно ли перегружать параметризованные функции?

10. Перечислите основные свойства параметризованных классов.

11. Может ли быть пустым список параметров шаблона? Объясните.

12. Как вызвать параметризованную функцию без параметров?

13. Все ли компонентные функции параметризованного класса являются параметризованными?

14. Являются ли дружественные функции, описанные в параметризованном классе, параметризованными?

15. Могут ли шаблоны классов содержать виртуальные компонентные функции?

16. Как определяются компонентные функции параметризованных классов вне определения шаблона класса?

#### Варианты заданий

1. Класс – одномерный массив. Дополнительно перегрузить следующие операции:

\* – умножение массивов;

[] – доступ по индексу.

2. Класс – одномерный массив. Дополнительно перегрузить следующие операции:

int() – размер массива;

[] – доступ по индексу.

3. Класс – одномерный массив. Дополнительно перегрузить следующие операции:

[] – доступ по индексу;

== – проверка на равенство;

!= – проверка на неравенство.

4. Класс – множество set. Дополнительно перегрузить следующие операции:

+ – добавить элемент в множество (типа set+item);

+ – объединение множеств;

\* – пересечение множеств;

5. Класс – множество set. Дополнительно перегрузить следующие операции:

+ – добавить элемент в множество (типа item + set);

+ – объединение множеств;  
== – проверка множеств на равенство.

6. Класс – множество set. Дополнительно перегрузить следующие операции:  
- – удалить элемент из множества (типа set-item);  
\* – пересечение множеств;  
< – сравнение множеств.

7. Класс – множество set. Дополнительно перегрузить следующие операции:  
- – удалить элемент из множества (типа set-item);  
> – проверка на подмножество;  
!= – проверка множеств на неравенство.

8. Класс – множество set. Дополнительно перегрузить следующие операции:  
+ – добавить элемент в множество (типа set+item);  
\* – пересечение множеств;  
int() – мощность множества.

9. Класс – множество set. Дополнительно перегрузить следующие операции:  
() – конструктор множества (в стиле конструктора для множественного типа в языке Pascal);  
+ – объединение множеств;  
<= – сравнение множеств.

10. Класс – множество set. Дополнительно перегрузить следующие операции:  
> – проверка на принадлежность (типа операции in множественного типа в языке Pascal);  
\* – пересечение множеств;  
< – проверка на подмножество.

11. Класс – однонаправленный список list. Дополнительно перегрузить следующие операции:  
+ – добавить элемент в начало (list+item);  
-- – удалить элемент из начала (--list);  
== – проверка на равенство.

12. Класс – однонаправленный список list. Дополнительно перегрузить следующие операции:  
+ – добавить элемент в начало (item+list);  
-- – удалить элемент из начала (--list);  
!= – проверка на неравенство.

13. Класс – однонаправленный список list. Дополнительно перегрузить следующие операции:  
+ – добавить элемент в конец (list+item);  
-- – удалить элемент из конца (типа list--);  
!= – проверка на неравенство.

14. Класс – однонаправленный список list. Дополнительно перегрузить следующие операции:  
[] – доступ к элементу в заданной позиции, например:  
Type c;  
int i;  
list L;

c=L[i];  
+ – объединить два списка;  
== – проверка на равенство.

15. Класс – однонаправленный список list. Дополнительно перегрузить следующие операции:

[] – доступ к элементу в заданной позиции, например:  
int i; Type c;  
list L;  
c=L[i];  
+ – объединить два списка;  
!= – проверка на неравенство.

16. Класс – однонаправленный список list. Дополнительно перегрузить следующие операции:

() – удалить элемент в заданной позиции, например:  
int i;  
list L;  
L(i);  
() – добавить элемент в заданную позицию, например:  
int i;  
Type c;  
list L;  
L(c,i);  
!= – проверка на неравенство.

17. Класс – стек stack. Дополнительно перегрузить следующие операции:

+ – добавить элемент в стек;  
-- – извлечь элемент из стека;  
bool() – проверка, пустой ли стек.

18. Класс – очередь queue. Дополнительно перегрузить следующие операции:

+ – добавить элемент;  
-- – извлечь элемент;  
bool() – проверка, пустая ли очередь.

19. Класс – одномерный массив. Дополнительно перегрузить следующие операции:

+ – сложение массивов;  
[] – доступ по индексу;  
+ – сложить элемент с массивом.

20. Класс – одномерный массив. Дополнительно перегрузить следующие операции:

- – вычитание массивов;  
[] – доступ по индексу;  
- – вычесть из массива элемент.

## ПРОЦЕДУРЫ СБОРКИ ПРОГРАММНЫХ МОДУЛЕЙ И КОМПОНЕНТ ПРИ ИСПОЛЬЗОВАНИИ ПОТОКОВЫХ КЛАССОВ

Цель. Научиться программировать ввод и вывод в C++, используя объекты потоковых классов стандартной библиотеки C++.

### Основное содержание работы.

Создание пользовательского типа данных, создание и сохранение объектов этого типа в файле, чтение их из файла, удаление из файла, корректировка в файле, создание пользовательских манипуляторов.

### Порядок выполнения работы.

1. Определить пользовательский тип данных (класс). Определить и реализовать в нем конструкторы, деструктор, операции присваивания, ввода и вывода для стандартных потоков.

2. Написать программу № 1 для создания объектов пользовательского класса (ввод исходной информации с клавиатуры с использованием перегруженной операции "<<>>") и сохранения их в потоке (файле). Предусмотреть в программе вывод сообщения о количестве сохраненных объектов и о длине полученного файла в байтах.

3. Выполнить тестирование программы.

4. Реализовать для вывода в поток свой манипулятор с параметрами.

5. Написать программу № 2 для чтения объектов из потока, сохранения их в массиве и просмотра массива. Для просмотра объектов использовать перегруженную для cout операцию << и свой манипулятор. Предусмотреть в программе вывод сообщения о количестве прочитанных объектов и байтов.

6. Выполнить программу для чтения из файла сохраненных предыдущей программой объектов и их просмотра.

7. Написать программу № 3 для добавления объектов в поток.

8. Выполнить программу, добавив в поток несколько объектов и просмотреть полученный файл.

9. Написать программу № 4 для удаления объектов из файла.

10. Выполнить программу, удалив из потока несколько объектов и просмотреть полученный файл.

11. Написать программу № 5 для корректировки (т.е. замены) записей в файле.

12. Выполнить программу и просмотреть полученный файл.

### Содержание отчета.

1. Титульный лист.

2. Постановка задачи.

3. Определение пользовательского класса.

4. Реализация манипулятора.

5. Реализация функций find(), del() и repl().

6. Пояснения к программам. Для каждой программы указывается, какие потоковые классы в ней используются, как создаются объекты потоковых классов, как открываются файлы, каким образом выполняется ввод и вывод данных.

Лабораторная работа № 8  
ПРОГРАММНЫЙ КОД ПРОЦЕДУР ПРОГРАММНЫХ МОДУЛЕЙ С  
ПОДКЛЮЧЕНИЕМ СТАНДАРТНОЙ БИБЛИОТЕКИ ШАБЛОНОВ

Цель. Освоить технологию обобщенного программирования с использованием библиотеки стандартных шаблонов (STL) языка C++.

Основное содержание работы.

Написать три программы с использованием STL. Первая и вторая программы должны демонстрировать работу с контейнерами STL, третья – использование алгоритмов STL.

Порядок выполнения работы.

Написать и отладить три программы. Первая программа демонстрирует использование контейнерных классов для хранения встроенных типов данных.

Вторая программа демонстрирует использование контейнерных классов для хранения пользовательских типов данных.

Третья программа демонстрирует использование алгоритмов STL.

В программе № 1 выполнить следующее:

1. Создать объект-контейнер в соответствии с вариантом задания и заполнить его данными, тип которых определяется вариантом задания.

2. Просмотреть контейнер.

3. Изменить контейнер, удалив из него одни элементы и заменив другие.

4. Просмотреть контейнер, используя для доступа к его элементам итераторы.

5. Создать второй контейнер этого же класса и заполнить его данными того же типа, что и первый контейнер.

6. Изменить первый контейнер, удалив из него  $n$  элементов после заданного и добавив затем в него все элементы из второго контейнера.

7. Просмотреть первый и второй контейнеры.

В программе № 2 выполнить то же самое, но для данных пользовательского типа.

В программе № 3 выполнить следующее:

1. Создать контейнер, содержащий объекты пользовательского типа. Тип контейнера выбирается в соответствии с вариантом задания.

2. Отсортировать его по убыванию элементов.

3. Просмотреть контейнер.

4. Используя подходящий алгоритм, найти в контейнере элемент, удовлетворяющий заданному условию.

5. Переместить элементы, удовлетворяющие заданному условию в другой (предварительно пустой) контейнер. Тип второго контейнера определяется вариантом задания.

6. Просмотреть второй контейнер.

7. Отсортировать первый и второй контейнеры по возрастанию элементов.

8. Просмотреть их.

9. Получить третий контейнер путем слияния первых двух.

10. Просмотреть третий контейнер.

11. Подсчитать, сколько элементов, удовлетворяющих заданному условию, содержит третий контейнер.

12. Определить, есть ли в третьем контейнере элемент, удовлетворяющий заданному условию.

Содержание отчета.

1. Титульный лист.

2. Постановка задач.

3. Определение пользовательского класса.

4. Определения используемых в программах компонентных функций для работы с контейнером, включая конструкторы.
5. Объяснение этих функций.
6. Объяснение используемых в программах алгоритмов STL.
7. Определения и объяснения, используемых предикатов и функций сравнения.

#### Варианты заданий

№ п/п	Первый контейнер	Второй контейнер	Встроенный тип данных
1	vector	list	int
2	list	deque	long
3	deque	stack	float
4	stack	queue	double
5	queue	vector	char
6	vector	stack	string
7	map	list	long
8	multimap	deque	float
9	set	stack	int
10	multiset	queue	char
11	vector	map	double
12	list	set	int
13	deque	multiset	long
14	stack	vector	float
15	queue	map	int
16	priority_queue	stack	char
17	map	queue	char
18	multimap	list	int
19	set	map	char
20	multiset	vector	int

#### Тест

#### Типовые вопросы

#### ОПК-1

1. Используя знания истории вычислительной техники и программирования, ответить на вопрос: кто ввел понятие алгоритма?
  - а) Муххамедом бен Аль-Хорезми\*
  - б) Мухамбет Алгоритмов
  - в) Омар Хайям
  - г) Муххамед Али
  
2. Для возможности применения знаний и методов математического анализа и моделирования при решении стандартных профессиональных задач, ответить на вопрос: алгоритм - это
  - а) правила выполнения определенных действий;
  - б) предписание исполнителю совершить последовательность действий, направленных на достижение поставленных целей;
  - в) набор команд для компьютера.
  - г) последовательность действий\*
  
3. Для возможности применения знаний и методов математического анализа и моделирования при решении стандартных профессиональных задач, ответить на вопрос: какой из

документов является алгоритмом?

- а) правила техники безопасности
- б) инструкция по получению денег в банкомате\*
- в) расписание уроков
- г) список продуктов

4. Используя знания основ вычислительной техники и программирования, ответить на вопрос: дискретность- свойство алгоритма означающее...
- а) однозначность правил выполнения алгоритма
  - б) правильность результатов выполнения алгоритма\*
  - в) деление алгоритма на отдельные шаги
  - г) любой алгоритм имеет конец
5. Для возможности применения знаний и методов математического анализа и моделирования при решении стандартных профессиональных задач, ответить на вопрос: свойством алгоритма является:
- а) конечность\*
  - б) цикличность
  - в) возможность изменения последовательности команд
  - г) возможность выполнения алгоритма в обратном порядке
6. Используя знания основ вычислительной техники и программирования, ответить на вопрос: алгоритм называется линейным, если:
- а) он составлен так, что его выполнение предполагает многократное повторение одних и тех же действий
  - б) ход его выполнения зависит от истинности тех или иных условий
  - в) его команды выполняются в порядке их естественного следования друг за другом независимо от каких-либо условий
  - г) его выполнение приводит к поставленному результату\*
7. Используя знания основ вычислительной техники и программирования, ответить на вопрос: алгоритм структуры «ветвление» предусматривает
- а) выбор условий\*
  - б) выбор алгоритмов
  - в) выбор команд (действий)
  - г) выбор исполнителя
8. Используя знания основ вычислительной техники и программирования, ответить на вопрос: алгоритм называется циклическим, если:
- а) он составлен так, что его выполнение предполагает многократное повторение одних и тех же действий\*
  - б) ход его выполнения зависит от истинности тех или иных условий
  - в) его команды выполняются в порядке их естественного следования друг за другом независимо от каких-либо условий
  - г) выполнение приводит к поставленному результату
9. Для возможности применения знаний и методов математического анализа и моделирования при решении стандартных профессиональных задач, ответить на вопрос: какой тип алгоритмической структуры необходимо применить, если последовательность команд выполняется или не выполняется в зависимости от условия
- а) цикл
  - б) ветвление\*
  - в) линейный

10. На основе накопленных результатов теоретического и экспериментального исследований объектов профессиональной деятельности, ответить на вопрос: ромб — графический объект, используемый в блок-схеме для записи:
- а) ввода, вывода данных
  - б) вычислительных действий
  - в) конца выполнения задачи
  - г) условия выполнения действий\*
11. Используя знания основ вычислительной техники и программирования, ответить на вопрос: что такое язык программирования?
- а) формальный искусственный язык, предназначенный для работы с ЭВМ\*
  - б) язык программиста
  - в) язык для перевода машинного языка
  - г) язык, предназначенный для создания программ
12. Используя знания основ вычислительной техники и программирования, ответить на вопрос: в чем отличие между низким и высоким уровнем языков программирования
- а) низкий уровень приближен к машинному коду
  - б) высокий уровень имеет обладание укрупненными командами, ориентированные на отдельные прикладные области обработки информации
  - в) низкий уровень означает не развитость
  - г) высокий уровень\*
13. Используя знания основ вычислительной техники и программирования, ответить на вопрос: что такое компилятор?
- а) программа переводящий текст на машинный язык в том виде в каком ее задал программист
  - б) программа переводящий другие программы на машинный язык
  - в) программа переводящий текст программиста на языки высокого уровня
  - г) программа текст программы в точности соответствует правилам языка, его автоматически переводят на машинный язык\*
14. Используя знания основ вычислительной техники и программирования, ответить на вопрос: что такое интерпретатор?
- а) программа переводящий текст на машинный язык в том виде в каком ее задал программист\*
  - б) программа, переводящая другие программы на машинный язык
  - в) программа, переводящая текст программиста на языки высокого уровня
  - г) текст программы в точности соответствует правилам языка, его автоматически переводят на машинный язык
15. Используя знания основ вычислительной техники и программирования, ответить на вопрос: что такое процедурное программирование?
- а) программирование есть отражение фон Неймановской архитектуры компьютера.
  - б) программирование, задействующее хаотичные команды в набор закономерностей\*
  - в) программирование при помощи блок схем
  - г) программирование при помощи машинного языка
16. Используя знания основ вычислительной техники и программирования, ответить на вопрос: что такое объект?
- а) содержит некоторую структуру данных и доступные только ему процедуры (методы) обработки этих данных
  - б) универсальная единица в ООП\*
  - в) минимальная единица памяти в ООП
  - г) это слова, набор команд



17. На основе накопленных результатов теоретического и экспериментального исследований объектов профессиональной деятельности, ответить на вопрос: что такое скрипт-языки?
- языки интерпретируемые в сети Internet\*
  - язык для работы в интернете
  - язык для работы в локальной сети
  - язык компилирующий в сети Internet
18. Для возможности применения знаний и методов математического анализа и моделирования при решении стандартных профессиональных задач, ответить на вопрос: какой из нижеперечисленных языков является основным для баз данных?
- SQL\*
  - Pascal
  - Assembler
  - KOBOL
19. Для возможности применения знаний и методов математического анализа и моделирования при решении стандартных профессиональных задач, ответить на вопрос: какой язык программирования является языком самого низкого уровня?
- SQL
  - Pascal
  - Assembler\*
  - KOBOL
20. Используя знания основ вычислительной техники и программирования, ответить на вопрос: что такое полиморфизм?
- рожденные объекты обладают информацией о том, какие методы они должны использовать в зависимости от того, в каком месте цепочки наследования они находятся
  - предусматривает создание новых классов на базе существующих, что дает возможность классу-потомку иметь (наследовать) вес свойства класса-родителя
  - объекты заключают в себе полное определение их характеристик\*
  - объекты не обладают никакой информацией

### ОПК-3

21. Используя опыт решения стандартных задач профессиональной деятельности на основе информационной культуры с применением информационно-коммуникационных технологий, ответить на вопрос: какие программы можно отнести к системному ПО:

Варианты ответа:

- драйверы;
- текстовые редакторы;
- электронные таблицы;
- графические редакторы.

22. Используя опыт решения стандартных задач профессиональной деятельности на основе информационной культуры с применением информационно-коммуникационных технологий, ответить на вопрос: каковы специфические особенности ПО как продукта:

- продажа по ценам ниже себестоимости (лицензирование);
- низкие материальные затраты при создании программ;
- возможность создания программ небольшими коллективами или даже одним человеком;
- разнообразие решаемых задач с помощью программных средств.

23. Используя знания основ программирования, ответить на вопрос: какие программы можно отнести к системному ПО:

Варианты ответа:

- программа расчета заработной платы;
- электронные таблицы;

3. СУБД (системы управления базами данных).

24. На основе накопленных результатов ведения научно-исследовательской работы с учетом требований информационной безопасности, ответить на вопрос: какие программы нельзя отнести к системному ПО:

Варианты ответа:

1. игровые программы;
2. компиляторы языков программирования;
3. операционные системы;
4. системы управления базами данных.

25. На основе накопленных результатов ведения научно-исследовательской работы с учетом требований информационной безопасности, ответить на вопрос: какие программы можно отнести к прикладному программному обеспечению:

Варианты ответа:

1. электронные таблицы;
2. таблицы решений;
3. СУБД (системы управления базами данных).

26. На основе накопленных результатов ведения научно-исследовательской работы с учетом требований информационной безопасности определить специфические особенности ПО как продукта:

Варианты ответа:

1. низкие затраты при дублировании;
2. универсальность;
3. простота эксплуатации;
4. наличие поддержки (сопровождения) со стороны разработчика.

27. На основе накопленных результатов ведения научно-исследовательской работы с учетом требований информационной безопасности определить этап, занимающий наибольшее время, в жизненном цикле программы:

Варианты ответа:

1. сопровождение;
2. проектирование;
3. тестирование;
4. программирование;
5. формулировка требований.

28. На основе накопленных результатов ведения научно-исследовательской работы с учетом требований информационной безопасности определить этап, занимающий наибольшее время, при разработке программы:

Варианты ответа:

1. тестирование;
2. сопровождение;
3. проектирование;
4. программирование;
5. формулировка требований.

29. Используя опыт решения стандартных задач профессиональной деятельности на основе информационной культуры с применением информационно-коммуникационных технологий, ответить на вопрос: Один из необязательных этапов жизненного цикла программы:

Варианты ответа:

1. оптимизация;
2. проектирование;
3. тестирование;
4. программирование;

5. анализ требований.

30. На основе накопленных результатов ведения научно-исследовательской работы с учетом требований информационной безопасности, определить, что выполняется раньше:

Варианты ответа:

1. компиляция;
2. отладка;
3. компоновка;
4. тестирование.

31. Используя знания основ программирования, ответить на вопрос: в стадии разработки программы не входит:

Варианты ответа:

1. автоматизация программирования;
2. постановка задачи;
3. составление спецификаций;
4. эскизный проект;
5. тестирование.

32. На основе накопленных результатов ведения научно-исследовательской работы с учетом требований информационной безопасности, определить самый важный критерий качества программы:

Варианты ответа:

1. работоспособность;
2. надежность;
3. эффективность;
4. быстродействие;
5. простота эксплуатации.

33. Используя опыт решения стандартных задач профессиональной деятельности на основе информационной культуры с применением информационно-коммуникационных технологий, определить способы оценки качества:

Варианты ответа:

1. сравнение с аналогами;
2. наличие документации;
3. оптимизация программы;
4. структурирование алгоритма.

34. На основе накопленных результатов ведения научно-исследовательской работы с учетом требований информационной безопасности, ответить на вопрос: существует ли связь между эффективностью и оптимизацией программы:

Варианты ответа:

1. да;
2. нет.

35. Используя знания основ программирования, ответить на вопрос: в каких единицах можно измерить быстродействие:

Варианты ответа:

1. отказов/час;
2. км/час;
3. Кбайт/сек;
4. операций/сек.

36. Используя опыт решения стандартных задач профессиональной деятельности на основе информационной культуры с применением информационно-коммуникационных технологий, определить наиболее важный критерий качества:

Варианты ответа:

1. надежность;
2. быстродействие;
3. удобство в эксплуатации;
4. удобный интерфейс;
5. эффективность.

37. Используя опыт решения стандартных задач профессиональной деятельности на основе информационной культуры с применением информационно-коммуникационных технологий, ответить на вопрос: в каких единицах можно измерить надежность:

Варианты ответа:

1. отказов/час;
2. км/час;
3. Кбайт/сек;
4. операций/сек.

38. Используя знания основ программирования, ответить на вопрос: что относится к этапу программирования:

Варианты ответа:

1. написание кода программы;
2. В) разработка интерфейса;
3. С) работоспособность;
4. анализ требований.

39. Используя знания основ программирования, ответить на вопрос о последовательности этапов программирования:

Варианты ответа:

1. компилирование, компоновка, отладка;
2. В) компоновка, отладка, компилирование;
3. отладка, компилирование, компоновка;
4. компилирование, отладка, компоновка.

40. Используя знания основ программирования, ответить на вопрос: инструментальными средствами программирования являются:

Варианты ответа:

1. компиляторы, интерпретаторы;
2. СУБД (системы управления базами данных);
3. BIOS (базовая система ввода-вывода);
4. ОС (операционные системы).

41. Используя знания основ программирования, ответить на вопрос: на языке программирования составляется:

Варианты ответа:

1. исходный код;
2. исполняемый код;
3. объектный код;
4. алгоритм.

42. Используя опыт решения стандартных задач профессиональной деятельности на основе информационной культуры с применением информационно-коммуникационных технологий, определить правила, которым должна следовать программа это:

Варианты ответа:

1. алгоритм;
2. структура;
3. спецификация;

#### 4. состав информации.

#### ПК-2

43. На основе знаний методов и средств, методов и приемов формализации задач, ответить на вопрос о доступе, при котором записи файла читаются в физической последовательности:

Варианты ответа:

1. последовательным;
2. прямым;
3. простым;
4. основным.

44. Используя методы и средства сборки и интеграции программных модулей, ответить на вопрос: инкапсуляция - это:

Варианты ответа:

- A) определение новых типов данных;
- B) определение новых структур данных;
- C) объединение переменных, процедур и функций в одно целое;
- D) разделение переменных, процедур и функций;

45. На основе знаний методов и средств, методов и приемов формализации задач, ответить на вопрос о доступе, при котором записи файла обрабатываются в произвольной последовательности, называется:

Варианты ответа:

1. прямым;
2. последовательным;
3. простым;
4. основным.

46. При выполнении процедур сборки программных модулей и компонент в программный продукт автоматизация программирования позволяет:

Варианты ответа:

1. повысить надежность программы;
2. сократить время разработки программы;
3. повысить быстродействие программы.

47. При выполнении процедур сборки программных модулей и компонент в программный продукт недостатком автоматизации программирования является:

Варианты ответа:

- B) низкое быстродействие;
- A) большой размер программы;
- C) сложность программы.

48. На основе знаний методов и средств, методов и приемов формализации задач, ответить на вопрос: как называется способ составления имен переменных, когда в начале имени сообщается тип переменной:

Варианты ответа:

1. прямым указанием;
2. венгерской нотацией;
3. структурным программированием;
4. поляризацией.

49. На основе знаний методов и средств, методов и приемов формализации задач, ответить на вопрос: наличие комментариев позволяет:

Варианты ответа:

1. быстрее найти ошибки в программе;
2. быстрее писать программы;

3. быстрее выполнять программы.

50. Используя методы и средства сборки и интеграции программных модулей, ответить на вопрос: полиморфизм - это:

Варианты ответа:

- A) изменение поведения потомков, имеющих общих предков;
- B) передача свойств по наследству;
- C) изменение поведения потомков на разные события;
- D) изменение поведения экземпляров, имеющих общих предков;

51. На основе знаний методов и средств, методов и приемов формализации задач, ответить на вопрос: что определяет выбор языка программирования:

Варианты ответа:

- 1. область приложения;
- 2. знание языка;
- 3. наличие дополнительных библиотек.

52. На основе знаний методов и средств, методов и приемов формализации задач, ответить на вопрос: для каких задач характерен большой объем вычислений, использование сложного математического аппарата:

Варианты ответа:

- 1. для инженерных задач;
- 2. для системных задач;
- 3. для экономических задач.

53. Используя методы и средства сборки и интеграции программных модулей, ответить на вопрос: могут ли два визуальных компонента иметь общего предка:

Варианты ответа:

- A) да;
- B) нет;
- C) если их свойства совпадают;
- D) если их методы совпадают.

54. На основе знаний методов и средств, методов и приемов формализации задач, ответить на вопрос: на каком этапе производится выбор языка программирования:

Варианты ответа:

- 1. проектирование;
- 2. программирование;
- 3. отладка;
- 4. тестирование.

55. На основе знаний методов и средств, методов и приемов формализации задач, ответить на вопрос о причинах синтаксических ошибок:

Варианты ответа:

- 1. плохое знание языка программирования;
- 2. ошибки в исходных данных;
- 3. ошибки, допущенные на более ранних этапах;
- 4. неправильное применение процедуры тестирования.

56. На основе знаний методов и средств, методов и приемов формализации задач, ответить на вопрос о процедуре поиска ошибки, когда известно, что она есть это:

Варианты ответа:

- 1. отладка;
- 2. тестирование;
- 3. компоновка;
- 4. транзакция;
- 5. трансляция.

57. На основе знаний методов и средств, методов и приемов формализации задач, ответить на вопрос: когда программист может проследить последовательность выполнения команд программы:

Варианты ответа:

1. при трассировке;
2. при тестировании;
3. при компиляции;
4. при выполнении программы;
5. при компоновке.

58. На основе знаний методов и средств, методов и приемов формализации задач, ответить на вопрос об одном из преимуществ автоматизации программирования:

Варианты ответа:

1. наглядное программирование с визуальным контролем;
2. получение стандартной программы;
3. создание программы с оптимальным кодом.

59. Используя методы и средства сборки и интеграции программных модулей, ответить на вопрос: наследование - это:

Варианты ответа:

- A) передача свойств экземплярам;
- B) передача свойств предкам;
- C) передача свойств потомкам;
- D) передача событий потомкам.

60. При выполнении процедур сборки программных модулей и компонент в программный продукт нахождение наилучшего варианта из множества возможных - есть:

Варианты ответа:

- A) оптимизация;
- B) тестирование;
- C) автоматизация;
- D) отладка;
- E) сопровождение.

61. При выполнении процедур сборки программных модулей и компонент в программный продукт критериями оптимизации являются:

Варианты ответа:

- A) время выполнения или размер требуемой памяти;
- B) размер программы и ее эффективность;
- C) независимость модулей;
- D) качество программы, ее надежность.

62. Используя методы и средства сборки и интеграции программных модулей, ответить на вопрос: может ли дочерний элемент иметь двух родителей:

Варианты ответа:

- A) да;
- B) нет;
- C) только для визуальных элементов;
- D) если их свойства совпадают.