

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ
федеральное государственное автономное образовательное учреждение высшего
образования
"САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ"

Кафедра № 42

УТВЕРЖДАЮ
Ответственный за образовательную
программу

доц. д.ф.н., доц.

(должность, уч. степень, звание)

П.М. Колычев

(подпись, фамилия)

(подпись)
«20» 02 2025 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

«Основы программирования»
(Наименование дисциплины)

Код направления подготовки/ специальности	51.03.01
Наименование направления подготовки/ специальности	Культурология
Наименование направленности	Цифровая культура и цифровое искусство
Форма обучения	очная
Год приема	2025

Лист согласования рабочей программы дисциплины

Программу составил (а)

доцент, к.т.н., доцент
(должность, уч. степень, звание)

 06.02.2025
(подпись, дата)

Н.В. Богословская
(инициалы, фамилия)

Программа одобрена на заседании кафедры № 42

«06» 02 2025 г, протокол № 6/2024-25

Заведующий кафедрой № 42

д.т.н., доц.
(уч. степень, звание)

 06.02.25
(подпись, дата)

С.В. Мичурин
(инициалы, фамилия)

Заместитель декана факультета №6 по методической работе

проф. д.и.н., доц.
(должность, уч. степень, звание)

 10.02.25
(подпись, дата)

Л.Ю. Гусман
(инициалы, фамилия)

Аннотация

Дисциплина «Основы программирования» входит в образовательную программу высшего образования – программу бакалавриата по направлению подготовки/специальности 51.03.01 «Культурология» направленности «Цифровая культура и цифровое искусство». Дисциплина реализуется кафедрой «№42».

Дисциплина нацелена на формирование у выпускника следующих компетенций:

ПК-4 «Способен разрабатывать техническую документацию на продукцию в сфере информационных технологий, управления технической информацией, которая изложена в соответствии с возможностями и потребностями в сфере цифровой культуры и цифрового искусства»

Содержание дисциплины охватывает круг вопросов, связанных с основами объектно-ориентированной концепции: инкапсуляция, наследование, полиморфизм, библиотеки классов. Изучаются базовые принципы объектного подхода, который в последующих учебных дисциплинах будет использован в объектно-ориентированном анализе, объектно-ориентированном проектировании, шаблонах проектирования, объектно-ориентированных данных (XML), унифицированном языке моделирования Unified Modeling Language (UML), объектно-ориентированной разработке и конструировании программных систем.

В практической части дисциплины формируются навыки объектного программирования, отладки и тестирования прототипов программных приложений. По всем ключевым темам дисциплины разработаны задания для выполнения практических и лабораторных работ. Основная цель курсовой работы - использование полученных знаний и навыков программирования для разработки собственной библиотеки классов, расширяющей функциональность стандартных библиотек используемой платформы.

Преподавание дисциплины предусматривает следующие формы организации учебного процесса: лекции, лабораторные работы, практические занятия, самостоятельная работа обучающегося, курсовое проектирование. Программой дисциплины предусмотрены следующие виды контроля: текущий контроль успеваемости, промежуточная аттестация в форме экзамена.

Общая трудоемкость освоения дисциплины составляет 6 зачетных единиц, 216 часов.

Язык обучения по дисциплине «русский»

1. Перечень планируемых результатов обучения по дисциплине

1.1. Цели преподавания дисциплины— освоение студентами фундаментальных концепций объектно-ориентированного программирования как общеинженерного знания для теоретического и экспериментального исследования в профессиональной деятельности. Получение практического опыта использования инструментария объектно-ориентированной разработки для решения практических задач в области информационных технологий.

1.2. Дисциплина входит в состав части, формируемой участниками образовательных отношений, образовательной программы высшего образования (далее – ОП ВО).

1.3. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения ОП ВО.

В результате изучения дисциплины обучающийся должен обладать следующими компетенциями или их частями. Компетенции и индикаторы их достижения приведены в таблице 1.

Таблица 1 – Перечень компетенций и индикаторов их достижения

Категория (группа) компетенции	Код и наименование компетенции	Код и наименование индикатора достижения компетенции
Профессиональные компетенции	ПК-4 Способен разрабатывать техническую документацию на продукцию в сфере информационных технологий, управления технической информацией, которая изложена в соответствии с возможностями и потребностями в сфере цифровой культуры и цифрового искусства	ПК-4.3.2 знать архитектурные решения, применяемые при проектировании программных средств и компьютерных систем различного назначения; стандарты в области системной и программной инженерии ПК-4.У.1 уметь компоновать документ на основе заданных источников; подготавливать графические схемы; описывать бизнес-процессы в сфере цифровой культуры и цифрового искусства с помощью графических нотаций ПК-4.У.3 уметь в сфере цифровой культуры и цифрового искусства разрабатывать требования к техническому документу и к комплекту технической документации; разрабатывать технические задания и спецификации требований; составлять календарный план выполнения полученного задания; разрабатывать описание системной или программной архитектуры; разрабатывать руководства пользователя

2. Место дисциплины в структуре ОП

Дисциплина может базироваться на знаниях, ранее приобретенных обучающимися при изучении следующих дисциплин:

– «Информатика».

Знания, полученные при изучении материала данной дисциплины, имеют как самостоятельное значение, так и используются при изучении других дисциплин:

– «Алгоритмы и структуры данных»,

– «Трехмерное моделирование».

3. Объем и трудоемкость дисциплины

Данные об общем объеме дисциплины, трудоемкости отдельных видов учебной работы по дисциплине (и распределение этой трудоемкости по семестрам) представлены в таблице 2.

Таблица 2 – Объем и трудоемкость дисциплины

Вид учебной работы	Всего	Трудоемкость по семестрам	
		№4	№5
1	2	3	4
Общая трудоемкость дисциплины, ЗЕ/ (час)	6/ 216	3/ 108	3/ 108
Из них часов практической подготовки	68	34	34
Аудиторные занятия, всего час.	102	51	51
в том числе:			
лекции (Л), (час)	34	17	17
практические/семинарские занятия (ПЗ), (час)			
лабораторные работы (ЛР), (час)	68	34	34
курсовой проект (работа) (КП, КР), (час)			
экзамен, (час)	36		36
Самостоятельная работа, всего (час)	78	57	21
Вид промежуточной аттестации: зачет, дифф. зачет, экзамен (Зачет, Дифф. зач, Экз.**))	Зачет, Экз.	Зачет	Экз.

Примечание: ** кандидатский экзамен

4. Содержание дисциплины

4.1. Распределение трудоемкости дисциплины по разделам и видам занятий.

Разделы, темы дисциплины и их трудоемкость приведены в таблице 3.

Таблица 3 – Разделы, темы дисциплины, их трудоемкость

Разделы, темы дисциплины	Лекции (час)	ПЗ (СЗ) (час)	ЛР (час)	КП (час)	СРС (час)
Семестр 4					
Раздел 1. Основы языка C/C++ Тема 1.1. Типы данных. Преобразование типов Тема 1.2. Структура программы. Библиотеки. Компилятор Тема 1.3 Синтаксис и операторы языка C	3		8		17
Раздел 2. Инструменты языка C/C++ Тема 2.1. Функции. Макросы Тема 2.2. Указатель, ссылки Тема 2.3. Массивы, статические и динамические. Выделение памяти. Тема 2.4. Символы, строки, хеширование Тема 2.5. Рекурсивные функции Тема 2.6 Структуры Тема 2.7. Поразрядные операции	10		16		20

Раздел 3. Анализ и оценка сложности алгоритмов Тема 3.1. Асимптотическая оценка сложности алгоритмов. Классы сложности. Тема 3.2. Оценка сложности операций доступа и поиска данных. Тема 3.3. Сортировка пузырьком. Сортировка расческой. Гномья сортировка Тема 3.4. Сортировка вставками. Сортировка Шелла	4		8		20
Итого в семестре:	17		34		57
Семестр 5					
Раздел 4. Архитектура классов	10		20		11
Раздел 5. Основные отношения между классами в UML	7		14		10
Итого в семестре:	17		34		21
Итого	34	0	68	0	78

Практическая подготовка заключается в непосредственном выполнении обучающимися определенных трудовых функций, связанных с будущей профессиональной деятельностью.

4.2. Содержание разделов и тем лекционных занятий.

Содержание разделов и тем лекционных занятий приведено в таблице 4.

Таблица 4 – Содержание разделов и тем лекционного цикла

Номер раздела	Название и содержание разделов и тем лекционных занятий
1	<p>Раздел 1. Основы языка C/C++</p> <p>Тема 1.1. Типы данных. Преобразование типов Представление данных в ЭВМ. Пользовательский тип данных. Двойственность типов данных. Преобразование типов данных, явное и неявное. Стандартные типы языка C/C++. Свойства типов данных.</p> <p>Тема 1.2. Структура программы. Библиотеки. Компилятор Подключение библиотек. Заголовки. Стандартные библиотеки. Функция main(). Глобальные и локальные переменные. Компиляция. Запуск и отладка программы.</p> <p>Тема 1.3 Синтаксис и операторы языка C Операторы языка. Бинарные, унарные операторы. Префиксные, постфиксные. Приоритет операций. Циклы с постусловием и предусловием for и while. Операторы прерывания break и continue. Условный оператор if. Оператор ветвления switch.</p>
2	<p>Раздел 2. Инструменты языка C/C++</p> <p>Тема 2.1. Функции. Макросы Понятие функции, прототипа функции. Перегрузка функций. Параметр по умолчанию. Возвращаемое значение функции. Макросы. Примеры использования функций и макросов. Вызов функции, пространство имен. Отладка работы функции.</p> <p>Тема 2.2. Указатель, ссылки Косвенная адресация. Переменные и указатели на них. Получение</p>

	<p>адреса. Операторы * и &.</p> <p>Тема 2.3. Массивы, статические и динамические. Выделение памяти. Понятие массива. Особенности выделения памяти для массива данных, статического и динамического. Способы обращения к элементу массива, указатели и массивы.</p> <p>Тема 2.4. Символы, строки, хеширование. Основные таблицы символов ASCII, CP-1251, UTF-8, UTF-16. Обработка строк с использованием <stdio>, <iostream>. Хеширование строк.</p> <p>Тема 2.5. Рекурсивные функции. Понятие рекурсии и рекурсивной функции. Условие завершения рекурсивного вызова, бесконечный вызов рекурсивной функции. Косвенная рекурсия. Глубина рекурсии.</p> <p>Тема 2.6 Структуры. Объявление структур в языке C. Доступ к элементам структуры. Выравнивание структур.</p> <p>Тема 2.7. Поразрядные операции. Поразрядные или побитовые операторы “и”&,”или” ,”не” ~, “исключающее или” ^, операторы сдвига >>,<<. Циклический сдвиг.</p>
3	<p>Раздел 3. Анализ и оценка сложности алгоритмов.</p> <p>Тема 3.1. Асимптотическая оценка сложности алгоритмов. Классы сложности.</p> <p>Тема 3.2. Оценка сложности операций доступа и поиска данных. Оценка времени доступа к данным и поиска для структур данных: массив, список, дерево.</p> <p>Тема 3.3. Сортировка пузырьком. Сортировка расческой. Гномья сортировка.</p> <p>Тема 3.4. Сортировка вставками. Сортировка Шелла.</p>
4	<p>Раздел 4. Архитектура классов.</p> <p>Тема 4.1 Базовые принципы объектно-ориентированного программирования: Инкапсуляция. Классы и объекты. Модификаторы доступа. Поля классов. Разработка библиотеки классов.</p> <p>Тема 4.2 Наследование классов. Реализация наследования: перегрузка и переопределение. Назначение конструкторов класса. Статические конструкторы.</p> <p>Тема 4.3. Свойства классов – механизм доступа к данным класса. Виртуальные свойства.</p> <p>Тема 4.4 Полиморфизм. Методы классов: перегрузка и переопределение. Виртуальные методы. Скрытие методов. Параметры методов.</p> <p>Тема 4.5 Иерархия классов. Абстрактные классы. Обобщенные типы. Класс System.Object. Назначение</p>

	абстрактных классов. Наследование обобщенных типов.
5	<p>Раздел 5. Основные отношения между классами в UML.</p> <p>Тема 5.1 События классов. Обработчики событий. Технология создания события и подписки на событие. Практические примеры использования событий классов.</p> <p>Тема 5.2 Интерфейсы – механизм множественного наследования в C#. Назначение интерфейсов. Технология разработки интерфейсов и подписки классов. Практические примеры использования интерфейсов для реализации множественного наследования.</p> <p>Тема 5.3. Отношения между классами и их реализация. Диаграмма классов в UML. Реализация отношений в структуре классов. Практические приемы использования отношений классов. Универсальные шаблоны</p>

4.3. Практические (семинарские) занятия

Темы практических занятий и их трудоемкость приведены в таблице 5.

Таблица 5 – Практические занятия и их трудоемкость

№ п/п	Темы практических занятий	Формы практических занятий	Трудоемкость, (час)	Из них практической подготовки, (час)	№ раздела дисциплины
Учебным планом не предусмотрено					
Всего					

4.4. Лабораторные занятия

Темы лабораторных занятий и их трудоемкость приведены в таблице 6.

Таблица 6 – Лабораторные занятия и их трудоемкость

№ п/п	Наименование лабораторных работ	Трудоемкость, (час)	Из них практической подготовки, (час)	№ раздела дисциплины
Семестр 4				
1	Алгоритм с использованием условного оператора	2	2	1
2	Алгоритм с использованием циклов	2	2	1
3	Одномерные массивы	4	4	1
4	Посимвольная обработка строк	2	2	2
5	Анализ строк и хеширование	4	4	2
6	Рекурсивный алгоритм	2	2	2
7	Алгоритм с использованием рекурсии и циклов	4	4	2
8	Бинарные (битовые) операции	2	2	2
9	Обработка битовых блоков и последовательностей	4	4	2
10	Способы измерения времени работы	4	4	3

	алгоритмов			
11	Оценка сложности алгоритмов	4	4	3
Семестр 5				
1	Классы и объекты. Поля и конструктор по умолчанию. Уровни доступа. Инициализаторы объектов. Проект библиотеки. Событийно-ориентированный интерфейс пользователя.	2	2	4
2	Конструкторы класса. Статический конструктор. Структуры.	2	2	4
3	Свойства класса.	2	2	4
4	Эргономика взаимодействия человек-система. Приемы построения интерфейса пользователя. Методы как интерфейс класса.	2	2	4
5	Источники данных для классов. Чтение, запись текстовых файлов. Параметры методов.	2	2	4
6	Механизмы наследования. Преобразования типов.	2	2	4
7	Полиморфизм. Виртуальные члены классов.	2	2	4
8	Иерархия классов. Абстрактные классы и члены класса.	2	2	4
9	Библиотека классов.	4	4	4
10	Классы и интерфейсы.	2	2	5
11	События классов. Событийно-ориентированный интерфейс пользователя.	4	4	5
12	Основные отношения между классами в UML: наследование, реализация, ассоциация, композиция, агрегация.	4	4	5
13	Разработка интерфейса программных систем. Графические объекты в приложении пользователя.	4	4	5
Всего		68	68	

4.5. Курсовое проектирование/ выполнение курсовой работы
Учебным планом не предусмотрено

4.6. Самостоятельная работа обучающихся
Виды самостоятельной работы и ее трудоемкость приведены в таблице 7.

Таблица 7 – Виды самостоятельной работы и ее трудоемкость

Вид самостоятельной работы	Всего, час	Семестр 4, час	Семестр 5, час
1	2	3	4
Изучение теоретического материала дисциплины (ТО)	27	20	7
Курсовое проектирование (КП, КР)			
Расчетно-графические задания (РГЗ)			
Выполнение реферата (Р)			

Подготовка к текущему контролю успеваемости (ТКУ)	24	17	7
Домашнее задание (ДЗ)			
Контрольные работы заочников (КРЗ)			
Подготовка к промежуточной аттестации (ПА)	27	20	7
Всего:	78	57	21

5. Перечень учебно-методического обеспечения
для самостоятельной работы обучающихся по дисциплине (модулю)
Учебно-методические материалы для самостоятельной работы обучающихся указаны в п.п. 7-11.

6. Перечень печатных и электронных учебных изданий
Перечень печатных и электронных учебных изданий приведен в таблице 8.
Таблица 8– Перечень печатных и электронных учебных изданий

Шифр/ URL адрес	Библиографическая ссылка	Количество экземпляров в библиотеке (кроме электронных экземпляров)
УДК 004.438 ББК 32.973.2018.1 Б90	Проектирование интерактивных приложений на языке С# : учебное пособие / Д. А. Булгаков ; С.-Петерб. гос. ун-т аэрокосм. приборостроения. - Электрон. текстовые дан. - Санкт-Петербург : Изд-во ГУАП, 2022. - 159 с.	
УДК 004.43(075) ББК 32.973.26-018.1 Ш96	Объектно-ориентированное программирование : [Электронный ресурс] : учебное пособие / Е. О. Шумова ; С.-Петерб. гос. ун-т аэрокосм. приборостроения. - Электрон. текстовые дан. - Санкт-Петербург : Изд-во ГУАП, 2021. - 115 с.	
УДК 004.45(075) ББК 32.972.1я73 М76	Молчанов, Алексей Юрьевич (канд. техн. наук, доц.). Разработка корпоративных программных решений на платформе .NET : учебное пособие / А. Ю. Молчанов, А. В. Аксенов ; С.-Петерб. гос. ун-т аэрокосм. приборостроения. - Электрон. текстовые дан. - Санкт-Петербург : Изд-во ГУАП, 2022. - 140 с	
http://lib.aanet.ru/	Основы программирования : [

	Электронный ресурс] : учебно-методическое пособие / сост.: Н. В. Богословская, А. В. Бржезовский. - Электрон. текстовые дан. - СПб. : [б. и.], 2019. - 95 с. - Систем. требования: ACROBAT READER 5.X. - Загл. с титул. экрана. - Б. ц.	
УДК 004.438 ББК 32.973.2018.1 Б90	Проектирование интерактивных приложений на языке C# : учебное пособие / Д. А. Булгаков ; С.-Петерб. гос. ун-т аэрокосм. приборостроения. - Электрон. текстовые дан. - Санкт-Петербург : Изд-во ГУАП, 2022. - 159 с.	
URL: https://urait.ru/bcode/562040	Трофимов, В. В. Алгоритмизация и программирование : учебник для вузов / В. В. Трофимов, Т. А. Павловская. — 4-е изд. — Москва : Издательство Юрайт, 2025. — 108 с. — (Высшее образование). — ISBN 978-5-534-20430-8. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/562040 (дата обращения: 22.01.2025).	
URL: https://urait.ru/bcode/563618	Огнева, М. В. Программирование на языке C++: практический курс : учебник для вузов / М. В. Огнева, Е. В. Кудрина, А. А. Казачкова. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 342 с. — (Высшее образование). — ISBN 978-5-534-18949-0. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/563618 (дата обращения: 22.01.2025).	

7. Перечень электронных образовательных ресурсов информационно-телекоммуникационной сети «Интернет»

Перечень электронных образовательных ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины приведен в таблице 9.

Таблица 9 – Перечень электронных образовательных ресурсов информационно-телекоммуникационной сети «Интернет»

№ п/п	Наименование
1.	https://docs.microsoft.com/ru-ru/dotnet/csharp/ - Docs / .NET / Руководство по языку C#
2.	https://znanium.com – ЭБС Знаниум
3.	https://urait.ru - ЭБС Юрайт
4.	https://e.lanbook.com – ЭБС Лань

8. Перечень информационных технологий

8.1. Перечень программного обеспечения, используемого при осуществлении образовательного процесса по дисциплине.

Перечень используемого программного обеспечения представлен в таблице 10.

Таблица 10– Перечень программного обеспечения

№ п/п	Наименование
	Microsoft Visual Studio/ Бесплатная версия на сайте Microsoft https://visualstudio.microsoft.com/ru/downloads/

8.2. Перечень информационно-справочных систем,используемых при осуществлении образовательного процесса по дисциплине

Перечень используемых информационно-справочных систем представлен в таблице 11.

Таблица 11– Перечень информационно-справочных систем

№ п/п	Наименование
	Библиотека MSDN — библиотека официальной технической документации для разработчиков под ОС Microsoft Windows. Режим доступа: https://docs.microsoft.com/ru-ru/

9. Материально-техническая база

Состав материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине, представлен в таблице12.

Таблица 12 – Состав материально-технической базы

№ п/п	Наименование составной части материально-технической базы	Номер аудитории (при необходимости)
1	Мультимедийная лекционная аудитория	
2	Компьютерный класс	

10. Оценочные средства для проведения промежуточной аттестации

10.1. Состав оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине приведен в таблице 13.

Таблица 13 – Состав оценочных средств для проведения промежуточной аттестации

Вид промежуточной аттестации	Перечень оценочных средств
------------------------------	----------------------------

Экзамен	Список вопросов к экзамену; Задачи; Тесты.
Зачет	Список вопросов; Тесты; Задачи.

10.2. В качестве критериев оценки уровня сформированности (освоения) компетенций обучающимися применяется 5-балльная шкала оценки сформированности компетенций, которая приведена в таблице 14. В течение семестра может использоваться 100-балльная шкала модульно-рейтинговой системы Университета, правила использования которой, установлены соответствующим локальным нормативным актом ГУАП.

Таблица 14 –Критерии оценки уровня сформированности компетенций

Оценка компетенции 5-балльная шкала	Характеристика сформированных компетенций
«отлично» «зачтено»	<ul style="list-style-type: none"> – обучающийся глубоко и всесторонне усвоил программный материал; – уверенно, логично, последовательно и грамотно его излагает; – опираясь на знания основной и дополнительной литературы, тесно привязывает усвоенные научные положения с практической деятельностью направления; – умело обосновывает и аргументирует выдвигаемые им идеи; – делает выводы и обобщения; – свободно владеет системой специализированных понятий.
«хорошо» «зачтено»	<ul style="list-style-type: none"> – обучающийся твердо усвоил программный материал, грамотно и по существу излагает его, опираясь на знания основной литературы; – не допускает существенных неточностей; – увязывает усвоенные знания с практической деятельностью направления; – аргументирует научные положения; – делает выводы и обобщения; – владеет системой специализированных понятий.
«удовлетворительно» «зачтено»	<ul style="list-style-type: none"> – обучающийся усвоил только основной программный материал, по существу излагает его, опираясь на знания только основной литературы; – допускает несущественные ошибки и неточности; – испытывает затруднения в практическом применении знаний направления; – слабо аргументирует научные положения; – затрудняется в формулировании выводов и обобщений; – частично владеет системой специализированных понятий.
«неудовлетворительно» «не зачтено»	<ul style="list-style-type: none"> – обучающийся не усвоил значительной части программного материала; – допускает существенные ошибки и неточности при рассмотрении проблем в конкретном направлении; – испытывает трудности в практическом применении знаний; – не может аргументировать научные положения; – не формулирует выводов и обобщений.

10.3. Типовые контрольные задания или иные материалы.

Вопросы (задачи) для экзамена представлены в таблице 15.

Таблица 15 – Вопросы (задачи) для экзамена

№ п/п	Перечень вопросов (задач) для экзамена	Код индикатора
1	Классы и структуры как основные конструкции системы общих типов CTS, используемой на платформе .NET.	ПК-4.3.2
2	Общая система типов платформы .NET (Common Type System – CTS). Основные конструкции CTS. Назначение класса Object.	ПК-4.3.2
3	Иерархия классов. Реализация наследования классов. Доступ к членам базового класса из класса-наследника.	ПК-4.3.2
4	Преобразование типов в иерархии классов. Скрытие свойств и методов класса.	ПК-4.3.2
5	Методы класса. Передача параметров методу. Статические методы и методы экземпляра.	ПК-4.3.2
6	Унаследованные и переопределенные методы. Перегрузка методов – частный случай полиморфизма. Скрытие производного класса членов базового класса.	ПК-4.3.2
7	Назначение и реализация интерфейса. Работа с объектами через интерфейсы. Интерфейсы и множественное наследование. Интерфейсы в преобразованиях типов.	ПК-4.3.2
8	Классы и объекты. Поля и конструктор по умолчанию. Инициализаторы объектов класса.	ПК-4.3.2
9	Уровни доступности типов и членов типов.	ПК-4.3.2
10	Конструкторы экземпляров и статические конструкторы. Статические классы.	ПК-4.3.2
11	Данные классов: поля как переменные класса и константные значения, связанные с классом. Модификаторы static, const, readonly.	ПК-4.3.2
12	Свойства класса. Свойства экземпляра, статические свойства. Автоматически реализуемые свойства. Переопределение свойств класса.	ПК-4.3.2
13	Унаследованные и переопределенные методы. Перегрузка методов – частный случай полиморфизма. Скрытие производного класса членов базового класса.	ПК-4.3.2
14	Преобразование типов в иерархии классов. Скрытие свойств и методов класса.	ПК-4.3.2
15	Основные отношения между классами: наследование, реализация. Одиночное и множественное наследование. Неявное наследование.	ПК-4.3.2
16	Основные отношения между классами: ассоциация, композиция и агрегация.	ПК-4.3.2
17	События классов. Определение и вызов событий. Класс данных события EventArgs.	ПК-4.3.2
18	<p>Имеется описание класса Rectangle</p> <pre>public class Rectangle { double length; double width; public Rectangle (double l, double w) { length = l; width = w; } public double getArea() { area = length*width; } }</pre>	ПК-4.У.3

	<pre>return (area); } } } Обеспечьте следующие условия: класс Circle также имеет метод getArea, изначально определенный как abstract в классе Shape. Напишите программный код классов.</pre>	
19	<p>Объекты классов Circle и Rectangle могут быть созданы следующими строками кода:</p> <pre>Circle circle = new Circle(5); Rectangle rectangle = new Rectangle(4,5);</pre> <p>Напишите код классов Circle и Rectangle.</p>	ПК-4.У.3
20	<p>Имеется описание класса Shape:</p> <pre>public abstract class Shape { protected double area; public abstract double calcArea(); }</pre> <p>Предложите вариант развития иерархии классов.</p>	ПК-4.У.3
21	<p>Имеется фрагмент описания класса</p> <pre>public class DataBaseReader { string dbName; int startPosition; public DataBaseReader (string name) { dbName = name; startPosition = 0;} public DataBaseReader (String name, int pos) { dbName = name; startPosition = pos;} }</pre> <p>Создайте объекты класса всеми возможными способами.</p>	ПК-4.У.3
22	Используя отношение композиции опишите структуру классов для следующей фразы «Руль – часть автомобиля».	ПК-4.У.2
23	Используя отношение агрегации опишите структуру классов для следующей фразы: «Компьютерная система — это целое. Компонентами являются монитор, клавиатура, мышь и системный блок компьютера».	ПК-4.У.2
24	Опишите структуру классов Employee и Child, с учетом того, что у сотрудника может быть любое количество детей.	ПК-4.У.2
25	Опишите структуру классов Employee и Department, с учетом того, что при определенных обстоятельствах работник может быть ассоциирован с несколькими отделами.	ПК-4.У.2
26	<p>Разработайте структуру класса Employee, для которого можно выполнить следующие действия:</p> <pre>Employee emp = new Employee(Convert.ToInt32(textBox2.Text), textBox1.Text); richTextBox1.Text += String.Format("Сотрудник {0} {2} с таб номером {1}\n", Employee.Enterprise, emp.EmpId, emp.Name); this.BackColor = Employee.BackColor; this.Font = Employee.FontText;</pre>	ПК-4.У.3

Вопросы (задачи) для зачета / дифф. зачета представлены в таблице 16.

Таблица 16 – Вопросы (задачи) для зачета / дифф. зачета

№ п/п	Перечень вопросов (задач) для зачета / дифф. зачета	Код индикатора
1	Структура программного кода, компиляция, файл кода программы.	ПК-4.3.2
2	Библиотеки. Заголовки. Подключение библиотек и заголовочных файлов.	ПК-4.3.2
3	Синтаксис C++. Операторы языка. Бинарные, унарные операторы. Префиксные, постфиксные.	ПК-4.3.2
4	Условный оператор. Особенности значений выражения условного оператора.	ПК-4.У.3
5	Циклы, операторы прерывания.	ПК-4.У.3
6	Переменные. Типы данных: статическая и динамическая типизация в разных языках программирования.	ПК-4.У.3
7	Типы данных. Преобразование типов, явное, неявное.	ПК-4.У.3
8	Типы данных. Пользовательские типы данных. Массивы.	ПК-4.У.3
9	Доступ к элементу массива, поиск элемента массива. Бинарный поиск в отсортированном массиве.	ПК-4.3.2
10	Указатели, ссылки. Принципы доступа к данным.	ПК-4.3.2
11	Ссылки и указатели в качестве аргумента функции.	ПК-4.3.2
12	Пустой указатель, преобразование указателя.	ПК-4.3.2
13	Динамическое выделение памяти.	ПК-4.3.2
14	Выделение памяти для многомерного массива.	ПК-4.3.2
15	Представление области памяти в виде двумерных и трехмерных массивов.	ПК-4.3.2
16	Структуры. Динамические структуры. Стеки, списки, деревья.	ПК-4.3.2
17	Двойное представление данных. Бинарные операции.	ПК-4.3.2
18	Простейшие алгоритмы с использованием бинарных операций.	ПК-4.У.2
19	Побитовые операции над типами с плавающей запятой.	ПК-4.3.2
20	Символы. Таблицы кодировок символов.	ПК-4.3.2
21	Строки. Представление строковых переменных.	ПК-4.3.2
22	Хэширование.	ПК-4.3.2
23	Потоковый ввод/вывод.	ПК-4.3.2
24	Форматированный ввод/вывод. Вывод на консоль и в файл.	ПК-4.3.2
25	Функции. Прототип функции. Параметр по умолчанию.	ПК-4.У.3
26	Перегрузка функций.	ПК-4.У.3
27	Рекурсия. Рекурсивные функции.	ПК-4.У.3
28	Встраиваемые функции. Макросы.	ПК-4.У.3
29	Асимптотическая оценка сложности алгоритмов.	ПК-4.У.1
30	Оценка сложности алгоритма. Оценка сложности циклов, процедуры поиска и доступа к элементу.	ПК-4.У.1
31	Оценка сложности алгоритма. Измерение времени работы алгоритма.	ПК-4.У.1
32	Задача сортировки массива. Сортировки: гномья, вставками, пузырьком, Шелла, расческой.	ПК-4.3.2
33	<p>Чему равно значение “*c” после выполнения команд?</p> <pre>int b[8] = { 7, 2, 11, 1, 2, 11, 8, 0 }; int* c;</pre>	ПК-4.У.3

	<pre> for (c = b; *c < 10; c++) *(c + 1) = *c + *(c + 1); </pre>	
34	<p>Что выведет на экран данный код?</p> <pre> int Compare(int A, int B) { if (A > B) printf("%d is greater than %d; ", A, B); else printf("%d is greater than %d; ", B, A); return A > B; } int main() { if (Compare(3, 7) Compare(7, 3)) printf("inevitability; "); } </pre>	ПК-4.У.3
35	Сколько аргументов может быть при вызове функции int F(float A[4], float B[4], float C)?	ПК-4.У.3
36	<p>Что возвращает функция</p> <pre> int F(int A, int B) { if (A > B) if (B > 100) B = 100; else B = A; return A + B; } </pre> <p>при вызове F(230, 90);</p>	ПК-4.У.3
37	Какой код выводит на экран все нечетные числа от 19 до 3, каждое число с новой строки?	ПК-4.У.3

Перечень тем для курсового проектирования/выполнения курсовой работы представлены в таблице 17.

Таблица 17 – Перечень тем для курсового проектирования/выполнения курсовой работы

№ п/п	Примерный перечень тем для курсового проектирования/выполнения курсовой работы
	Учебным планом не предусмотрено

Вопросы для проведения промежуточной аттестации в виде тестирования представлены в таблице 18.

Таблица 18 – Примерный перечень вопросов для тестов

№ п/п	Примерный перечень вопросов для тестов	Код индикатора
1	<p><i>Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа.</i></p> <p>Чем отличается объектно-ориентированный стиль программирования от структурного?</p> <p>1 В структурном программировании программа представляется как как система объектов, взаимодействие которых способно решить ту или иную задачу. В ООП на первый план выходит логика, понимание последовательности выполнения действий для</p>	ПК-4

	<p>достижения поставленной цели.</p> <p>2 В структурном программировании на первый план выходит логика, понимание последовательности выполнения действий для достижения поставленной цели. В ООП – важнее представить программу как как систему объектов, взаимодействие которых способно решить ту или иную задачу.</p> <p>3 В структурном программировании на первый план выходит возможность повторного использования кода. В ООП лежат принципы последовательной декомпозиции объектов, взаимодействие которых способно решить ту или иную задачу.</p> <p>4 Структурное программирование основано на модели построения программы как иерархии процедур. Для ООП основным принципом является отсутствие явного описания последовательности действий для выполнения программы.</p>							
2	<p><i>Прочитайте текст, выберите правильные варианты ответа и запишите аргументы, обосновывающие выбор ответов.</i></p> <p>Ключевыми принципами объектно-ориентированного программирования являются:</p> <p>1 Абстракция. Моделирование требуемых атрибутов и взаимодействий сущностей в виде классов для определения абстрактного представления системы.</p> <p>2 Инкапсуляция. Скрытие внутреннего состояния и функций объекта и предоставление доступа только через открытый набор функций.</p> <p>3 Наследование. Возможность создания новых абстракций на основе существующих.</p> <p>4 Полиморфизм. Возможность реализации наследуемых свойств или методов отличающимися способами в рамках множества абстракций.</p> <p>5 Формализация. Отображение результатов мышления в точных понятиях и утверждениях.</p>	ПК-4						
3	<p><i>Прочитайте текст и установите соответствие. К каждой позиции, данной в левом столбце, подберите соответствующую позицию в правом столбце.</i></p> <table><tr><td>1. Система программирования — это</td><td>А. комплекс программных средств, предназначенных для поддержки программного продукта на протяжении всего жизненного цикла этого продукта</td></tr><tr><td>2. Система — это</td><td>В. совокупность взаимосвязанных элементов, объединенных в одно целое для достижения некоторой цели, которая определяется назначением системы</td></tr><tr><td>3. Системное моделирование — это</td><td>междисциплинарное исследование использования моделей для концептуализации и построения систем в бизнесе и ИТ-разработке.</td></tr></table>	1. Система программирования — это	А. комплекс программных средств, предназначенных для поддержки программного продукта на протяжении всего жизненного цикла этого продукта	2. Система — это	В. совокупность взаимосвязанных элементов, объединенных в одно целое для достижения некоторой цели, которая определяется назначением системы	3. Системное моделирование — это	междисциплинарное исследование использования моделей для концептуализации и построения систем в бизнесе и ИТ-разработке.	ПК-4
1. Система программирования — это	А. комплекс программных средств, предназначенных для поддержки программного продукта на протяжении всего жизненного цикла этого продукта							
2. Система — это	В. совокупность взаимосвязанных элементов, объединенных в одно целое для достижения некоторой цели, которая определяется назначением системы							
3. Системное моделирование — это	междисциплинарное исследование использования моделей для концептуализации и построения систем в бизнесе и ИТ-разработке.							
4	<p><i>Прочитайте текст и установите последовательность. Запишите</i></p>	ПК-4						

	<i>соответствующую последовательность букв слева направо.</i> В зависимости от вида, масштабов и потребностей программного проекта определяется порядок разработки: А. Разработка и анализ требований В. Проектирование С. Дизайн D. Кодирование Е. Тестирование F. Внедрение G. Сопровождение							
5	<i>Прочитайте текст и запишите развернутый обоснованный ответ.</i> Какие парадигмы и стили программирования поддерживает C#	ПК-4						
11	<i>Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа.</i> Применение математических алгоритмов в оптимизации и анализе производительности программного кода помогает разработчикам и инженерам 1.сократить время выполнения программ, оптимизировать использование памяти и ресурсов, а также снизить энергопотребление. 2.оптимизировать использование человеческих (трудовых) ресурсов, а также снизить энергопотребление 3.снизить затраты на оплату труда персонала компании 4. ускорить выполнение бизнес-процессов компании	ПК-4						
12	<i>Прочитайте текст, выберите правильные варианты ответа и запишите аргументы, обосновывающие выбор ответов.</i> В программировании используются следующие методы математического анализа и моделирования: 1. Методы линейного программирования. 2. Градиентные методы. 3. Эволюционное моделирование основано на принципах естественного отбора и дарвиновской эволюции, где генетические алгоритмы применяются для генерации новых вариантов кода и отбора лучших. 4. Экспериментальное моделирование.	ПК-4						
13	<i>Прочитайте текст и установите соответствие. К каждой позиции, данной в левом столбце, подберите соответствующую позицию в правом столбце.</i> <table><tr><td>1. Методы линейного программирования позволяют</td><td>А. находить оптимальные значения переменных и ограничений, оптимизируя характеристики кода, такие как время выполнения или потребление памяти</td></tr><tr><td>2.Градиентные методы, такие как градиентный спуск, используются</td><td>В. нахождения локальных оптимумов и оптимизации функций производительности кода</td></tr><tr><td>3. Эволюционное моделирование основано на принципах</td><td>С. естественного отбора и дарвиновской эволюции, где генетические алгоритмы</td></tr></table>	1. Методы линейного программирования позволяют	А. находить оптимальные значения переменных и ограничений, оптимизируя характеристики кода, такие как время выполнения или потребление памяти	2.Градиентные методы, такие как градиентный спуск, используются	В. нахождения локальных оптимумов и оптимизации функций производительности кода	3. Эволюционное моделирование основано на принципах	С. естественного отбора и дарвиновской эволюции, где генетические алгоритмы	ПК-4
1. Методы линейного программирования позволяют	А. находить оптимальные значения переменных и ограничений, оптимизируя характеристики кода, такие как время выполнения или потребление памяти							
2.Градиентные методы, такие как градиентный спуск, используются	В. нахождения локальных оптимумов и оптимизации функций производительности кода							
3. Эволюционное моделирование основано на принципах	С. естественного отбора и дарвиновской эволюции, где генетические алгоритмы							

		применяются для генерации новых вариантов кода и отбора лучших.									
14	<p>Прочитайте текст и установите последовательность. Запишите соответствующую последовательность букв слева направо.</p> <p>Несмотря на появление новых тенденций, основные этапы разработки ПО остались неизменными:</p> <ul style="list-style-type: none">A. Описание целевого программного продукта;B. Проектирование продукта;C. Разработка продукта;D. Тестирование частей;E. Интеграция частей и тестирование продукта в целом;F. Сопровождение продукта.		ПК-4								
15	<p>Прочитайте текст и запишите развернутый обоснованный ответ.</p> <p>Программист является важным и востребованным специалистом в IT-сфере, обладающим уникальными навыками и знаниями в создании и разработке программного обеспечения. Сформулируете основные роли и обязанности программиста в современном информационном обществе.</p>		ПК-4								
16	<p>Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа.</p> <p>Что означает $O(N)$ для алгоритма?</p> <ul style="list-style-type: none">1. Алгоритм не имеет асимптотическое ограничение сверху2. Алгоритм имеет линейный рост сложности3. Время работы алгоритма зависит только от параметра N и не от чего больше4. Алгоритм имеет постоянную сложность равную некоторой константе		ПК-4								
17	<p>Прочитайте текст, выберите правильные варианты ответа и запишите аргументы, обосновывающие выбор ответов.</p> <p>Алгоритм бинарного поиска числа в структуре размерности N соответствует следующим утверждениям:</p> <ul style="list-style-type: none">1. Может быть реализован при помощи рекурсивной функции2. При применении к неотсортированному массиву выполняется за $O(N)$3. Является примером жадного алгоритма4. При работе с однонаправленным списком не выполняется гарантированно за $O(\log N)$5. Если искомое число является максимумом, то позволяет найти его в массиве за $O(1)$		ПК-4								
18	<p>Прочитайте текст и установите соответствие. К каждой позиции, данной в левом столбце, подберите соответствующую позицию в правом столбце.</p> <table><tr><td>$O(2^N)$</td><td>При различных исходных данных время работы алгоритма не превышает некоторое значение</td></tr><tr><td>$O(N)$</td><td>Позволяет предположить удвоение времени работы при увеличении размерности данных на 1</td></tr><tr><td>$O(1)$</td><td>Говорит о необходимости линейного увеличения затрачиваемых ресурсов при аналогичном росте размерности данных</td></tr><tr><td>$O(N^2)$</td><td>Асимптотическая сложность алгоритма сортировки вставками для</td></tr></table>		$O(2^N)$	При различных исходных данных время работы алгоритма не превышает некоторое значение	$O(N)$	Позволяет предположить удвоение времени работы при увеличении размерности данных на 1	$O(1)$	Говорит о необходимости линейного увеличения затрачиваемых ресурсов при аналогичном росте размерности данных	$O(N^2)$	Асимптотическая сложность алгоритма сортировки вставками для	ПК-4
$O(2^N)$	При различных исходных данных время работы алгоритма не превышает некоторое значение										
$O(N)$	Позволяет предположить удвоение времени работы при увеличении размерности данных на 1										
$O(1)$	Говорит о необходимости линейного увеличения затрачиваемых ресурсов при аналогичном росте размерности данных										
$O(N^2)$	Асимптотическая сложность алгоритма сортировки вставками для										

		среднего случая		
19	<p><i>Прочитайте текст и установите последовательность. Запишите соответствующую последовательность букв слева направо.</i></p> <p>Приведите классы сложности в порядке убывания скорости роста (сначала тот, что обладает наименьшей скоростью роста при $N \rightarrow \infty$)</p> <p>A. $(N^2 \log N)$ B. $(N \log^2 N)$ C. $(N^2 \log^2 N)$ D. $(N \log^3 N)$</p>	ПК-4		
20	<p><i>Прочитайте текст и запишите развернутый обоснованный ответ.</i></p> <p>Поиск общего элемента в K массивах размерности N. Пусть в каждой паре массивов содержится только один общий элемент. За какое минимальное асимптотическое время в зависимости от N и K можно найти общий для всех массивов элемент в случае, если массивы никак не упорядочены?</p>	ПК-4		
21	<p><i>Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа.</i></p> <p>Сколько раз выполняется преобразование типов в следующем коде C++?</p> <pre>int A = 10; float B = (int) A / 10; bool C = A > 10; if (A) B = C;</pre> <p>1. 2 2. 3 3. 4 4. 5 5. 0</p>	ПК-4		
21	<p><i>Прочитайте текст, выберите правильные варианты ответа и запишите аргументы, обосновывающие выбор ответов.</i></p> <p>Потоковый <code><iostream></code> ввод/вывод отличается от форматированного <code><stdio.h></code> ввода/вывода C++ тем, что:</p> <p>1. Потоковый ввод/вывод не позволяет форматировать данные для вывода 2. Наименованием библиотеки, в которой реализованы методы 3. Различным механизмом работы с классами 4. Форматированный ввод/вывод требует указания типа данных аргументов 5. Форматированный вывод <code><stdio.h></code> не позволяет выводить данные в буферный массив</p>	ПК-4		
23	<p><i>Прочитайте текст и установите соответствие. К каждой позиции, данной в левом столбце, подберите соответствующую позицию в правом столбце.</i></p> <p>Приведите в соответствие размеры структур языка C++, в байтах, 32 разрядная система</p> <table><tr><td>1. struct DataStr1 { float N; float Data[11]; short* ptr; };</td><td>A. 56</td></tr></table>	1. struct DataStr1 { float N; float Data[11]; short* ptr; };	A. 56	ПК-4
1. struct DataStr1 { float N; float Data[11]; short* ptr; };	A. 56			

	<pre> 2. struct DataStr2 { float Score; char Name[20]; char Surname[30]; }; 3. struct DataStr3 { double[6] Values; float Mean; DataStr3* next; }; </pre>	<p>B. 52</p> <p>C. 54</p>	
24	<p><i>Прочитайте текст и установите последовательность. Запишите соответствующую последовательность букв слева направо.</i></p> <p>Расположите следующие структуры данных по скорости доступа к элементу по его номеру в структуре:</p> <ul style="list-style-type: none"> A. Массив B. Однонаправленный список C. Дерево (хранение в массиве последовательно по уровням) D. Двухнаправленный список 		ПК-4
25	<p><i>Прочитайте текст и запишите развернутый обоснованный ответ.</i></p> <p>Что определяет данный код на языке C++ и для чего это может применяться:</p> <pre> struct MST { int N; float Data; MST* next; }; </pre>		ПК-4

Примечание:

Система оценивания тестовых заданий:

1 тип) Задание комбинированного типа с выбором одного верного ответа из четырех предложенных и обоснованием выбора считается верным, если правильно указана цифра и приведены конкретные аргументы, используемые при выборе ответа. Полное совпадение с верным ответом оценивается 1 баллом, неверный ответ или его отсутствие – 0 баллов.

2 тип) Задание комбинированного типа с выбором нескольких вариантов ответа из предложенных и развернутым обоснованием выбора считается верным, если правильно указаны цифры и приведены конкретные аргументы, используемые при выборе ответов. Полное совпадение с верным ответом оценивается 1 баллом, если допущены ошибки или ответ отсутствует – 0 баллов.

3 тип) Задание закрытого типа на установление соответствия считается верным, если установлены все соответствия (позиции из одного столбца верно сопоставлены с позициями другого столбца). Полное совпадение с верным ответом оценивается 1 баллом, неверный ответ или его отсутствие – 0 баллов.

4 тип) Задание закрытого типа на установление последовательности считается верным, если правильно указана вся последовательность цифр. Полное совпадение с верным ответом оценивается 1 баллом, если допущены ошибки или ответ отсутствует – 0 баллов.

5 тип) Задание открытого типа с развернутым ответом считается верным, если ответ совпадает с эталонным по содержанию и полноте. Правильный ответ за задание оценивается в 3 балла, если допущена одна ошибка \ неточность \ ответ

правильный, но не полный - 1 балл, если допущено более 1 ошибки \ ответ неправильный \ ответ отсутствует – 0 баллов.

Перечень тем контрольных работ по дисциплине обучающихся заочной формы обучения, представлены в таблице 19.

Таблица 19 – Перечень контрольных работ

№ п/п	Перечень контрольных работ
	Не предусмотрено

10.4. Методические материалы, определяющие процедуры оценивания индикаторов, характеризующих этапы формирования компетенций, содержатся в локальных нормативных актах ГУАП, регламентирующих порядок и процедуру проведения текущего контроля успеваемости и промежуточной аттестации обучающихся ГУАП.

11. Методические указания для обучающихся по освоению дисциплины

11.1. Методические указания для обучающихся по освоению лекционного материала.

Основное назначение лекционного материала – логически стройное, системное, глубокое и ясное изложение учебного материала. Назначение современной лекции в рамках дисциплины не в том, чтобы получить всю информацию по теме, а в освоении фундаментальных проблем дисциплины, методов научного познания, новейших достижений научной мысли. В учебном процессе лекция выполняет методологическую, организационную и информационную функции. Лекция раскрывает понятийный аппарат конкретной области знания, её проблемы, дает цельное представление о дисциплине, показывает взаимосвязь с другими дисциплинами.

Планируемые результаты при освоении обучающимися лекционного материала:

- получение современных, целостных, взаимосвязанных знаний, уровень которых определяется целевой установкой к каждой конкретной теме;
- получение опыта творческой работы совместно с преподавателем;
- развитие профессионально-деловых качеств, любви к предмету и самостоятельного творческого мышления.
- появление необходимого интереса, необходимого для самостоятельной работы;
- получение знаний о современном уровне развития науки и техники и о прогнозе их развития на ближайшие годы;
- научиться методически обрабатывать материал (выделять главные мысли и положения, приходить к конкретным выводам, повторять их в различных формулировках);
- получение точного понимания всех необходимых терминов и понятий.

Лекционный материал может сопровождаться демонстрацией слайдов и использованием раздаточного материала при проведении коротких дискуссий об особенностях применения отдельных тематик по дисциплине.

Структура предоставления лекционного материала:

- Вступительная часть определяет название темы, план и цель лекции.
- В основной части реализуется содержание темы, все главные узловые вопросы, рассматриваются приемы и технологии текущей темы с применением средств визуализации. Каждый вопрос заканчивается краткими выводами, логически подводящими студентов к следующему вопросу лекции.
- Заключительная часть обобщает в кратких формулировках основные идеи лекции, логически завершая ее как целостное рассмотрение темы.

11.2. Методические указания для обучающихся по выполнению лабораторных работ.

В ходе выполнения лабораторных работ обучающийся должен углубить и закрепить знания, практические навыки, овладеть современной методикой и техникой эксперимента в соответствии с квалификационной характеристикой обучающегося. Выполнение лабораторных работ состоит из экспериментально-практической, расчетно-аналитической частей и контрольных мероприятий.

Выполнение лабораторных работ обучающимся является неотъемлемой частью изучения дисциплины, определяемой учебным планом, и относится к средствам, обеспечивающим решение следующих основных задач обучающегося:

- приобретение навыков исследования процессов, явлений и объектов, изучаемых в рамках данной дисциплины;
- закрепление, развитие и детализация теоретических знаний, полученных на лекциях;
- получение новой информации по изучаемой дисциплине;
- приобретение навыков самостоятельной работы с лабораторным оборудованием и приборами.

Задание и требования к проведению лабораторных работ

Учебно-методическое пособие с заданиями к лабораторным работам имеется в электронной библиотеке ГУАП:

Основы программирования : [Электронный ресурс] : учебно-методическое пособие / сост.: Н. В. Богословская, А. В. Бржезовский. - Электрон. текстовые дан. - СПб. : [б. и.], 2019. - 95 с. - Систем. требования: ACROBAT READER 5.X. - Загл. с титул. экрана. - Б. ц.

Структура и форма отчета о лабораторной работе

Отчет о выполнении лабораторной работы должен включать следующие позиции: постановку задачи, описание исследуемой проблемы, пошаговое описание технологии выполнения с необходимыми комментариями к программному коду, результаты и выводы.

Требования к оформлению отчета о лабораторной работе

Изложение текста и оформление работы следует оформлять в соответствии с требованиями, представленными на сайте ГУАП <https://guap.ru/regdocs/docs/uch>

11.3. Методические указания для обучающихся по прохождению самостоятельной работы

В ходе выполнения самостоятельной работы, обучающийся выполняет работу по заданию и при методическом руководстве преподавателя, но без его непосредственного участия.

Для обучающихся по заочной форме обучения, самостоятельная работа может включать в себя контрольную работу.

В процессе выполнения самостоятельной работы, у обучающегося формируется целесообразное планирование рабочего времени, которое позволяет им развивать умения и навыки в усвоении и систематизации приобретаемых знаний, обеспечивает высокий уровень успеваемости в период обучения, помогает получить навыки повышения профессионального уровня.

Методическими материалами, направляющими самостоятельную работу обучающихся являются:

- учебно-методический материал по дисциплине;

– методические указания по выполнению контрольных работ (для обучающихся по заочной форме обучения).

11.4. Методические указания для обучающихся по прохождению текущего контроля успеваемости.

Текущий контроль успеваемости предусматривает контроль качества знаний обучающихся, осуществляемого в течение семестра с целью оценивания хода освоения дисциплины.

Учитываются посещения лекций и лабораторных занятий. Отчеты и практические результаты выполнения всех лабораторных работ должны быть защищены.

11.5. Методические указания для обучающихся по прохождению промежуточной аттестации.

Промежуточная аттестация обучающихся предусматривает оценивание промежуточных и окончательных результатов обучения по дисциплине. Она включает в себя:

– экзамен – форма оценки знаний, полученных обучающимся в процессе изучения всей дисциплины или ее части, навыков самостоятельной работы, способности применять их для решения практических задач. Экзамен, как правило, проводится в период экзаменационной сессии и завершается аттестационной оценкой «отлично», «хорошо», «удовлетворительно», «неудовлетворительно».

– дифференцированный зачет – это форма оценки знаний, полученных обучающимся при изучении дисциплины, при выполнении курсовых проектов, курсовых работ, научно-исследовательских работ и прохождении практик с аттестационной оценкой «отлично», «хорошо», «удовлетворительно», «неудовлетворительно».

Система оценок при проведении промежуточной аттестации осуществляется в соответствии с требованиями Положений «О текущем контроле успеваемости и промежуточной аттестации студентов ГУАП, обучающихся по программы высшего образования» и «О модульно-рейтинговой системе оценки качества учебной работы студентов в ГУАП».

Лист внесения изменений в рабочую программу дисциплины

Дата внесения изменений и дополнений. Подпись внесшего изменения	Содержание изменений и дополнений	Дата и № протокола заседания кафедры	Подпись зав. кафедрой