

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

Кафедра № 2

УТВЕРЖДАЮ

Ответственный за образовательную
программу

ДОЦ., К.Т.Н., ДОЦ.

(должность, уч. степень, звание)

В.А. Галанина

(инициалы, фамилия)



(подпись)

« 09 » _____ 02 _____ 2026 __ г

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

«Основы программирования»
(Наименование дисциплины)

Код направления подготовки/ специальности	09.03.03
Наименование направления подготовки/ специальности	Прикладная информатика
Наименование направленно- сти	Прикладная информатика и программирование
Форма обучения	очная

Санкт-Петербург– 2026__

Лист согласования рабочей программы дисциплины

Программу составил (а)

доцент, к.т.н., доцент

(должность, уч. степень, звание)



06.02.2026

(подпись, дата)

В.А.Галанина

(инициалы, фамилия)

Программа одобрена на заседании кафедры № 2

« 06 » _____ 02 _____ 2026 _____ г, протокол № 7/25-26 _____

Заведующий кафедрой № 2

д.ф.-м.н., проф.

(уч. степень, звание)



06.02.2026

(подпись, дата)

В.Г. Фарафонов

(инициалы, фамилия)

Заместитель директора института ФПТИ по методической работе

доц., к.т.н.

(должность, уч. степень, звание)



06.02.2026

(подпись, дата)

Н.Ю. Ефремов

(инициалы, фамилия)

Аннотация

Дисциплина «Основы программирования» входит в образовательную программу высшего образования – программу бакалавриата по направлению подготовки/ специальности 09.03.03 «Прикладная информатика» направленности «Прикладная информатика и программирование». Дисциплина реализуется кафедрой «№2».

Дисциплина нацелена на формирование у выпускника следующих компетенций:

УК-2 «Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений»

ОПК-2 «Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности»

ОПК-7 «Способен разрабатывать алгоритмы и программы, пригодные для практического применения»

ПК-1 «Способен выполнять сбор, систематизацию, выявление взаимосвязей и документирование требований к компьютерному программному обеспечению»

ПК-3 «Способен анализировать возможность реализации требований к прикладному программному обеспечению»

ПК-4 «Способен разрабатывать прикладное программное обеспечение»

ПК-7 «Способен разрабатывать базы данных»

ПК-9 «Способен руководить разработкой программного кода»

Содержание дисциплины охватывает круг вопросов, связанных с изучением принципов алгоритмизации решения прикладных задач и реализации этих алгоритмов с использованием языка программирования высокого уровня.

Преподавание дисциплины предусматривает следующие формы организации учебного процесса: лекции, лабораторные работы, практические занятия, курсовое проектирование, самостоятельная работа обучающегося.

Программой дисциплины предусмотрены следующие виды контроля: текущий контроль успеваемости, промежуточная аттестация в форме экзамена (2 семестр), экзамена (3 семестр).

Общая трудоемкость освоения дисциплины составляет 11 зачетных единиц, 396 часа.

Язык обучения по дисциплине «русский»

1. Перечень планируемых результатов обучения по дисциплине

1.1. Цели преподавания дисциплины

Целью преподавания дисциплины является получение обучающимися необходимых знаний по теории алгоритмов, практических навыков алгоритмизации вычислительных задач, умения реализовывать алгоритмы с использованием языка высокого уровня (на примере языка Си), знания основ объектно-ориентированного программирования (на примере языка C++)

1.2. Дисциплина входит в состав обязательной части 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
Универсальные компетенции	УК-2 Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений	УК-2.В.3 владеть навыками использования цифровых средств для решения поставленной задачи
Общепрофессиональные компетенции	ОПК-2 Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности	ОПК-2.3.1 знать современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности ОПК-2.У.1 уметь выбирать современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности ОПК-2.В.1 владеть навыками применения современных информационных технологий и программных средств, в том числе отечественного производства, при решении задач профессиональной деятельности
Общепрофессиональные компетенции	ОПК-7 Способен разрабатывать алгоритмы и программы, пригодные для практического применения	ОПК-7.3.1 знать основные языки программирования и работы с базами данных, операционные системы и оболочки, современные программные среды разработки информационных систем и технологий ОПК-7.У.1 уметь применять языки про-

		граммирования и работы с базами данных, современные программные среды разработки информационных систем и технологий для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ ОПК-7.В.1 владеть навыками программирования, отладки и тестирования прототипов программно-технических комплексов задач
Профессиональные компетенции	ПК-1 Способен выполнять сбор, систематизацию, выявление взаимосвязей и документирование требований к компьютерному программному обеспечению	ПК-1.3.3 знать основы классификации и кодирования информации
Профессиональные компетенции	ПК-3 Способен анализировать возможность реализации требований к прикладному программному обеспечению	ПК-3.3.3 знать методологию разработки программного обеспечения и технологию программирования ПК-3.У.1 уметь проводить анализ исполнения требований к программному обеспечению
Профессиональные компетенции	ПК-4 Способен разрабатывать прикладное программное обеспечение	ПК-4.3.2 знать типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке прикладного программного обеспечения ПК-4.У.1 уметь использовать существующие типовые решения и шаблоны разработки компьютерного программного обеспечения
Профессиональные компетенции	ПК-7 Способен разрабатывать базы данных	ПК-7.3.5 знать основы программирования ПК-7.3.7 знать современные структурные языки программирования ПК-7.В.1 владеть современными объектно-ориентированными языками программирования
Профессиональные компетенции	ПК-9 Способен руководить разработкой программного кода	ПК-9.3.1 знать методы и приемы алгоритмизации поставленных задач ПК-9.3.4 знать технологии программирования ПК-9.У.1 уметь писать программный код на выбранном языке программирования

2. Место дисциплины в структуре ОП

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

– «Информатика» изучается в 1 семестре.

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

– «Архитектура программных систем и паттерны программирования» изучается в 4 и 5 семестрах;

– «Системное программирование» изучается в 4 семестре;

– «Вычислительные системы, сети и телекомплексы» изучается в 5 семестре;

– «Программная инженерия» изучается в 6 семестре;

– «Проектирование информационных систем» изучается в 7 семестре;

– «Методы машинного обучения» изучается в 8 семестре.

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	
		№2	№3
1	2	3	4
Общая трудоемкость дисциплины, ЗЕ/ (час)	11/ 396	6/ 216	5/ 180
Из них часов практической подготовки	75	43	32
Аудиторные занятия, всего час.	187	102	85
в том числе:			
лекции (Л), (час)	68	34	34
практические/семинарские занятия (ПЗ), (час)	34	34	
лабораторные работы (ЛР), (час)	68	34	34
курсовой проект (работа) (КП, КР), (час)	17		17
экзамен, (час)	90	54	36
Самостоятельная работа, всего (час)	119	60	59
Вид промежуточной аттестации: зачет, дифф. зачет, экзамен (Зачет, Дифф. зач, Экз.)	Экз., Экз., Курс. Раб.	Экз.,	Экз., Курс. Раб.

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 2					
Раздел 1. Современные информационных технологий и программные средства.					20
Тема 1.1. Современные методы разработки	1				

программных средств. Тема 1.2. Этапы разработки программных средств. Раздел 2. Основы алгоритмизации и программирования Тема 21.1. Понятие алгоритма. Свойства алгоритмов. Формы записей алгоритмов. ГОСТ на изображение алгоритмов	1				
Тема 2.2. Базовые и дополнительные управляющие структуры	2	8			
Тема 2.3. Реализация базовых и дополнительных управляющих структур на алгоритмических языках.	2	10			40
Раздел 3. Основы языка Си.	1	16			
Тема 3.1. История развития языка Си. Стандарты языка.	4				
Тема 3.2. Типы данных, операции, выражения	4				
Тема 3.3. Реализация потоков управления	3		6		
Тема 3.4. Типы переменных в языке C/C++	4				
Тема 3.5. Функции и структура программы на Си	4		8		
Тема 3.6. Указатели и массивы	4		8		
Тема 3.7. Форматный и потоковый ввод/вывод	4		4		
Тема 3.8. Работа с файлами	2		8		
Итого в семестре:	34	34	34		60
Семестр 3					
Раздел 4. Основы объектно-ориентированного программирования Тема 4.1. Объекты и классы. Тема 4.2. Работа с классами в C++ Тема 4.3. Перегрузка операторов в C++. Тема 4.4. Типы связей между объектами Тема 4.5. Наследование в C++ Тема 4.6. Шаблоны в C++ и контейнерные классы	6 6 5 5 6 6		8 10 8 8		29
Выполнение курсовой работы				17	30
Итого в семестре:	34		34	17	59

Итого	68	34	68	17	119

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.Современные информационных технологии и программные средства.</p> <p>Тема 1.1. Современные методы разработки программных средств. Развитие современных методов программирования. Структурное, процедурное, модульное и объектно-ориентированное программирование.</p> <p>Тема 1.2. Этапы разработки программных средств. Семь этапов разработки программ.</p>
2.	<p>Раздел 1. Основы алгоритмизации и программирования.</p> <p>Тема 1.1.Понятие и свойства алгоритмов. Понятие алгоритма. Свойства алгоритмов. Формы записей алгоритмов. Общие принципы построения алгоритмов.</p> <p>Тема 1.2. Базовые и дополнительные управляющие структуры</p> <p>Базовые управляющие структуры: линейный процесс, ветвящийся процесс (развилка), циклический процесс (цикл с предусловием). Дополнительные управляющие структуры: обход, выбор, цикл с постусловием, цикл с параметром.</p>
3.	<p>Раздел 2. Основы языка Си.</p> <p>Тема 2.1. История развития языка Си. Стандарты языка. История языка Си. Стандарты языка. Структура программы на языке Си.</p> <p>Тема 2.2. Типы данных, операции, выражения</p> <p>Базовые типы данных. Операции, выражения, операторы. Преобразования типов. Составные типы данных: массивы, строки, структуры, ограничения, перечисления.</p> <p>Тема 2.3. Реализация потоков управления</p> <p>Управляющие операторы: операторы while, for, do while, Операторы ветвления: if...else, оператор выбора switch.</p> <p>Тема 2.4. Типы переменных в языке C/C++</p> <p>Локальные и глобальные переменные. Время жизни и область видимости. Автоматические и статические переменные. Схемы хранения. Стековая, статическая и динамическая память</p> <p>Тема 2.5. Функции и структура программы на Си</p> <p>Использование функций. Аргументы функций. Перегрузка функций. Передача по значению и использование указателей. Использование ссылок. Создание прототипов функций..</p> <p>Тема 2.6. Указатели и массивы</p> <p>Массивы и указатели. Указатели и многомерные массивы. Динамические массивы. Функция malloc(). Использование new.</p> <p>Тема 2.7. Форматный и потоковый ввод/вывод.</p> <p>Функции printf() и scanf(). Использование объектов cin и cout для вво-</p>

	да/вывода данных. Тема 2.8. Работа с файлами Файловый ввод/вывод. Текстовые и двоичные файлы.
4.	Раздел 3. Основы объектно-ориентированного программирования. Тема 3.1. Объекты и классы История развития ООП. Базовые понятия ООП: объект, его свойства и методы, класс, интерфейс. Основные принципы ООП: инкапсуляция, наследование, полиморфизм. Структура класса. Спецификаторы доступа public и private. Конструкторы класса. Делегирующие конструкторы. Сеттеры, геттеры.. Деструкторы. Тема 3.2. Работа с классами в C++ Классы и заголовочные файлы. Статические переменные -члены класса. Дружественные функции и классы. Тема 3.3. Перегрузка операторов в C++. Перегрузка операторов через дружественные функции, методы класса и обычные функции. Примеры перегрузки операторов. Конструктор копирования. Поверхностное и глубокое копирование. Тема 3.4. Типы связей между объектами. Композиция объектов, агрегация, ассоциация. Тема 3. 5 Наследование в C++ Наследование классов: открытое, закрытое и защищенное. Управление доступом. Множественное наследование. Переопределение методов родительского класса. Абстрактные базовые классы. Виртуальные функции и классы. Тема 3.6. Шаблоны в C++и контейнерные классы Шаблоны функций и классов. Явная и частичная спецификация шаблона. Контейнерные классы. Классы vector, list.

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)	№ раз-дела дисциплины
Семестр 2					
1	Составление блок-схем линейных алгоритмов	Разбор задач под руководством преподавателя, самостоятельное решение задач.	8	3	2
2	Составление блок-схем ветвящихся алгоритмов		8	3	2
3.	Составление блок-схем циклических алгоритмов		10	3	2
4	Составление блок-схем алгоритмов сортировки данных		8	4	2
Всего			34	13	

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 2				
1	Работа со статическими массивами в языке Си	4	5	3
2	Использование указателей в языке Си. Обработка двумерных массивов.	6	5	3
3	Использование функций в языке Си.	6	5	3
4	Обработка строковых переменных. Класс string (введение в язык C++).	6	5	3
5	Динамические массивы. Программирование задач линейной алгебры.	6	5	
6	Работа с составными типами данных. Структуры	6	5	3
	Всего:	34	30	
Семестр 3				
1	Работа с текстовыми файлами. Класс fstream.	4	5	4
2	Работа с бинарными файлами. Разработка базы данных средствами C++	6	5	4
3	Классы в C++. Работа с графической библиотекой SMFT	6	5	4
4	Работа с классами. Наследование	6	5	4
5	Работа с классами. Абстрактные базовые классы.	6	5	4
6	Разработка контейнерного класса Vector	6	5	4
	Всего:	34	20	
	Всего	68		

4.5. Курсовое проектирование/ выполнение курсовой работы

Цель курсовой работы:

Закрепление знаний, полученных на лекциях и лабораторных работах путем применения их для решения конкретной практической задачи

Часов практической подготовки: 12

Примерные темы заданий на курсовую работу приведены в разделе 10 РПД.

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 2, час	Семестр 3, час
1	2	3	4
Изучение теоретического материала дисциплины (ТО)	17	40	27
Курсовое проектирование (КП, КР)	12		12
Подготовка к текущему контролю	15	10	10
Подготовка к промежуточной аттестации (ПА)	20	10	10
Всего:	119	60	59

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)
URL адрес	Наименование электронного учебного издания	
https://codelibs.ru/iazyk-c/?ysclid=mp3xt1jlm6999170520 <i>Режим доступа: для авторизованных пользователей.</i>	Брайан У. Керниган, Деннис М. Ритчи. Язык программирования С. Издание 2-е. М: «Диалектика», 2020.	
https://codelibs.ru/iazyk-programmirovaniia-c-dlia-professionalov-4-izd/?ysclid=mp3xzxtlih325212838 <i>Режим доступа: для авторизованных пользователей.</i>	Бьерн Страуструп. Язык программирования C++ для профессионалов. Издание 4-е. М: . 2025	
<i>Шифр в библиотеке ГУАП</i>	<i>Наименование печатного учебного издания</i>	<i>Количество экземпляров</i>
Г15	В.А.Галанина. С.Л.Козенко,	50

	<i>Основы программирования, ч.1/Учебно-методическое пособие – СПб, ГУАП, 2022. – 51 с.</i>	
Г15	<i>В.А.Галанина.С.Л.Козенко, Основы программирования», ч..2/ Учебно-методическое пособие – СПб, ГУАП, 2023. –60с.</i>	50
Г15	<i>В.А.Галанина.С.Л.Козенко. Основы объектно-ориентированного программирования - СПб, ГУАП, 2025. –50 с.</i>	50

**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»

URL адрес	Наименование
https://pro.guap.ru/	Элементы электронного курса (методические указания к выполнению лабораторных и курсовой работ) по дисциплине размещены внутри ЭИОС ГУАП «Интегрированная среда обучения»

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
	Программные средства общего назначения
1	Электронная информационно-образовательная среда ГУАП «Интегрированная среда обучения» (https://pro.guap.ru/) разработана сотрудниками ГУАП (введена в эксплуатацию приказом ГУАП от 06.06.2017 № 05-215/17), перечень модулей и их функциональное назначение изложены по ссылке https://guap.ru/it/system/iso
2	Официальный сайт образовательной организации в сети «Интернет»

	(https://guap.ru/), разработан сотрудниками ГУАП (введен в эксплуатацию Приказом ГУАП от 23.03.2023 № 05-145/23)
3	Microsoft Office 2019 (договор ГУАП, информация о лицензии представлена по ссылке https://guap.ru/it/system/iso/po)
	Специальные программные средства
	Интегрированная среда программирования Visual Studio Communication (распространяется без лицензии)

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
	Электронные библиотечные ресурсы и системы
1	Электронный каталог библиотеки ГУАП с доступом к базе полнотекстовых изданий (https://lib.guap.ru/), доступ через личный кабинет читателя библиотеки ГУАП
2	Научная электронная библиотека «eLIBRARY» (https://elibrary.ru/), доступ через личный кабинет читателя библиотеки ГУАП, а также по IP -адресам ГУАП
3	ЭБС «Лань» (https://e.lanbook.com/), доступ через личный кабинет читателя библиотеки ГУАП, а также по IP -адресам ГУАП
4	ЭБС Консорциума аэрокосмических вузов России (http://elsau.ru/suai), доступ по IP-адресам ГУАП
5	ЭБС Znanium (https://znanium.ru/), доступ через личный кабинет читателя библиотеки ГУАП, а также по IP -адресам ГУАП
6	образовательная платформа «Юрайт» (https://urait.ru/), доступ через личный кабинет читателя библиотеки ГУАП, а также по IP -адресам ГУАП
7	Научная электронная библиотека «КиберЛенинка» (https://cyberleninka.ru/), свободный доступ
	Информационные и справочно-правовые системы
1	"Консультант Плюс" (www.consultant.ru) сетевая версия для образовательных организаций, доступ по IP -адресам ГУАП

9. Материально-техническая база

Состав материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине, представлен в таблице 12.

Таблица 12 – Состав материально-технической базы

№ п/п	Наименование составной части материально-технической базы	Номер аудитории (при необходимости)
-------	---	-------------------------------------

1	Учебная аудитория для занятий лекционного типа. Мультимедийная лекционная аудитория: Специализированная мебель; технические средства обучения, служащие для представления учебной информации большой аудитории; набор демонстрационного оборудования (Интерактивный мультисенсорный дисплей на перекатной стойке FocusTouch Диагональ 70" – 1 шт., ПЭВМ – 1 шт.); Обеспечен доступ в электронную информационно-образовательную среду ГУАП по локальной вычислительной сети	ул.Гастелло, 15
2	Учебная аудитория для практических ,лабораторных работ, самостоятельной работы, для групповых и индивидуальных консультаций, для текущего контроля и промежуточной аттестации. Специализированная мебель; лабораторное оборудование: ПЭВМ - 23 шт., объединенных в локальную вычислительную сеть с выходом в вычислительную сеть ГУАП и Интернет; проектор подвесной EPSON EMP-X5e; экран ScreenMedia GoldView 183*244 MW настенный; сплиттер Kramer VP-200K (с блоком питания)	ул. Гастелло, 15 22-10
	Помещение для самостоятельной работы обучающихся - Читальный зал библиотеки ГУАП: специализированная мебель; персональные компьютеры – 10 шт., обеспечен доступ в электронную информационно-образовательную среду ГУАП по локальной вычислительной сети и точке доступа WiFi, а также к электронно-библиотечным системам, реферативной базе данных Scopus; копировальный аппарат Kyocera KM2035.	ул. Гастелло, 15

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;
Выполнение курсовой работы	Экспертная оценка на основе требований к содержанию курсовой работы по дисциплине.

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	Характеристика сформированных компетенций
5-балльная шкала	

Оценка компетенции	Характеристика сформированных компетенций
5-балльная шкала	
«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	1. Опишите, что такое современные информационные технологии в программировании? 2. Назовите 3–5 отечественных программных средств для разработки ПО. В чём их особенности? 3. Перечислите, какие задачи в профессиональной деятельности программиста могут решаться с помощью отечественного ПО? 4. Перечислите преимущества и недостатки у облачных технологий по сравнению с локальными решениями? 5. Назовите современные языки программирования, применяющиеся для решения профессиональных задач? 6. Приведите пример задачи, где использование отечественного	ОПК-2.3.1

	ПО будет более эффективным, чем зарубежного.	
2	<p>1. Назовите 3–5 основных языков программирования и укажите их области применения.</p> <p>2. Опишите, в чём различия между компилируемыми и интерпретируемыми языками программирования? Приведите примеры.</p> <p>3. Перечислите, какие языки программирования используются для работы с базами данных? Приведите примеры.</p> <p>4. Назовите языки программирования, которые используются для разработки мобильных приложений?</p> <p>5. Вспомните, какие языки программирования могут быть использованы для работы с базами данных.</p> <p>6. Опишите, какую роль в работе компьютера играет операционная система.</p>	ОПК-7.3.1
3	<p>1. Перечислите, какие методы сбора требований к программному обеспечению вы знаете?</p> <p>2. Назовите основные источники требований к ПО.</p> <p>3. Опишите, какие вопросы вы будете задать заказчику, чтобы выявить функциональные требования к ПО?</p> <p>4. Опишите, что такое нефункциональные требования к ПО? Приведите примеры.</p> <p>5. Перечислите, какие документы используются для фиксации требований к ПО?</p> <p>6. Перечислите, какие стандарты документирования требований вы знаете?</p>	ПК-1.3.3
4	<p>1. Назовите основные принципы разработки программ.</p> <p>2. Опишите семь этапов программирования</p> <p>3. Проиллюстрируйте, как нужно использовать ГОСТ на оформление программной документации</p> <p>4. Назовите принципы структурного программирования</p> <p>5. Перечислите базовые управляющие структуры</p> <p>6. Перечислите дополнительные управляющие структуры</p>	ПК-3.3.3
5	<p>1. Объясните, какие существуют способы алгоритмизации циклических вычислительных процессов</p> <p>2. Опишите структуру программ на языке C</p> <p>3. Перечислите основные элементы языка C</p> <p>4. Перечислите типы данных в языке C/C++</p> <p>5. Объясните, в чем отличие целочисленных типов без знака</p> <p>6. Объясните, верно ли, что символьные типы относятся к целочисленным типам?</p>	ПК-4.3.2
6	<p>1. Опишите базовые типы данных в языке C/C++.</p> <p>2. Перечислите целочисленные типы</p> <p>3. Расскажите, какие типы данных служат для представления вещественных чисел</p> <p>4. Назовите основные правила работы с константами с плавающей точкой</p> <p>5. Объясните разницу между преобразованием типов и приведением типов</p> <p>6. Опишите основные особенности использования массивов в качестве аргументов функций</p>	ПК-7.3.5
7	1. Объясните, чем отличается динамическое выделение памяти от	ПК-7.3.7

	<p>статического</p> <p>2. Объясните, для чего необходимо преобразование типов</p> <p>3. Перечислите составные типы данных в языке C/C++.</p> <p>4. Опишите, что такое композиция? Чем отличается композиция от агрегации</p> <p>5. Перечислите, когда вызывается конструктор копирования.</p> <p>6. Назовите, чем отличается метод от функции?</p>	
8	<p>1. Объясните, зачем нужны приватные члены в классе?</p> <p>2. Перечислите методы доступа к элементам класса</p> <p>3. Объясните понятие и назначение наследования</p> <p>4. Опишите, чем отличается конструктор от других методов класса</p> <p>5. Опишите, в каких случаях используется ключевое слово this и что оно обозначает ?</p> <p>6. Опишите, в чем отличие копирования от присваивания?</p>	ПК-9.3.1.
9	<p>1. Назовите, в каких случаях необходимо использовать глубокое копирование?</p> <p>2. Опишите, зачем нужны виртуальные функции.</p> <p>3. Опишите, как виртуальные функции влияют на размер класса.</p> <p>4. Определите понятие «дружественные методы класса»</p> <p>5. Опишите, в чем разница между определением и объявлением шаблона?</p> <p>6. Перечислите, в каком порядке вызываются конструкторы и деструкторы класса при создании и удалении объекта производного класса?</p>	ПК-9.3.4.
10	<p>1. Приведите пример задачи, для решения которой целесообразно использовать отечественное ПО. Обоснуйте выбор.</p> <p>2. Предложите, как можно применить облачные технологии в профессиональной деятельности программиста?</p> <p>3. Какие инструменты (в том числе отечественные) можно использовать для организации командной работы над программным проектом? Сравните их по функционалу и стоимости."</p> <p>4. Какие отечественные операционные системы (например, Astra Linux, Alt Linux) могут заменить Windows в корпоративной среде? Сравните их по поддерживаемому ПО и безопасности.</p> <p>5. Предложите 2-3 подходящих инструмента (включая отечественные) для разработки мобильного приложения с ограниченным бюджетом и обоснуйте их выбор.</p> <p>6. Приведите пример отечественных решений для кибербезопасности (антивирусы, SIEM-системы) В чём их особенности?</p>	ОПК-2.У.1.
11	<p>1. Приведите пример решения задач с использованием алгоритма ветвящейся структуры</p> <p>2. Приведите пример решения задач с использованием алгоритма циклической структуры</p> <p>3. Объясните, какие проблемы могут возникнуть при смешивании цифрового и строкового ввода</p> <p>4. Продемонстрируйте на примере, чем отличается динамическое выделение памяти в C от C++</p> <p>5. Как вы думаете, верно ли утверждение, что размер структуры</p>	ОПК-7.У.1

	<p>можно вычислить, как сумму размеров ее полей</p> <p>6. Опишите основные особенности использования структур в функциях</p>	
	<ol style="list-style-type: none"> 1. Покажите на примере, чем отличается использование указателей и ссылок в функциях C++ 2. Объясните связь схемы хранения и типа связывания. 3. Объясните на примере, каким образом осуществляется перегрузка функций и зачем она нужна 4. Продемонстрируйте на примере способы выделения динамической памяти под двумерные массивы в C++ 5. Приведите пример инициализации значений закрытых переменных в классе 6. Как вы думаете, можно ли утверждать, что в однофайловых программах глобальные переменные идентичны статическим с внутренним связыванием 	ПК-3.У.1
13	<ol style="list-style-type: none"> 1. Напишите программу, которая создаёт объект класса Student с атрибутами name и grade и выводит информацию о студенте. 2. Создайте класс Rectangle с атрибутами width и height и методом для вычисления площади. 3. Продемонстрируйте на примере, чем удобно использование класса string для работы со строками. 4. Как вы думаете, если производный класс не добавляет члены данных в базовый класс, то нужны ли конструкторы для производного класса? 5. Объясните, как работает механизм исключений в C++ 6. Как вы думаете, может ли шаблонный конструктор быть конструктором без параметров? 	ПК-4.У.1
14.	<ol style="list-style-type: none"> 1. Напишите программу на Си, которая вычисляет факториал числа, введённого пользователем 2. Используйте цикл for, чтобы вывести на экран все чётные числа от 1 до 100. 3. Примените условный оператор if-else, чтобы определить, является ли введённое число положительным, отрицательным или нулем. 4. Напишите функцию, которая принимает строку и возвращает её в обратном порядке 5. Напишите функцию, которая проверяет, является ли число простым. 6. Напишите программу, которая считывает данные из текстового файла и выводит их на экран построчно 	ПК-9.У.1
15	<ol style="list-style-type: none"> 1. Проанализируйте, какие цифровые инструменты (например, Excel, Python с Pandas, Power BI, отечественные аналоги) вы бы использовали, если нужно проанализировать большой набор данных о продажах компании? Опишите шаги анализа." 2. Оцените, какие цифровые инструменты (например, Figma, Pixonic Design, Astra Linux) вы бы применили для разработки прототипа мобильного приложения 3. Проанализируйте, какой цифровой инструмент лучше подой- 	УК-2.В.3

	<p>дет для разработки теста для проверки знаний студентов</p> <p>4. Проиллюстрируйте на примере, как вы отладите программу на C++ с использованием цифровых средств Visual Studio</p> <p>5. Проиллюстрируйте на примере, как вы протестируете корректность работы функции в Visual Studio</p> <p>6. Проиллюстрируйте на примере, как вы отслеживаете состояние стека при отладке программы на C++</p>	
16	<p>1. Проанализируйте предложенный код и скажите,какой результат будет получен</p> <pre>int main() { a=7; b=-10; c=a-b; std::cout<<"a="<<a; std::cout<<"b="<<d; std::cout<<"c="<<c; }</pre> <p>2. Проанализируйте предложенный код и скажите, какие проблемы возникают при объявлении дружественной функции для класса-шаблона? Как исправить ошибку?</p> <pre>template <class T> class MyClass { T m_value_; public: MyClass(const T &value) : m_value(value) {}; ... friend void fun(...); }; template <class T> void fun(...) {...};</pre> <p>3. Проанализируйте код программы и ответьте, какой результат будет получен?</p> <pre>int main() { char ch = 'D'; ch = ch + 1; std::cout << ch << ' ' << (int)ch; }</pre> <p>4.Проанализируйте фрагмент кода и скажите, есть ли в нем ошибка</p> <pre>... int f= 53; char g={f}; ...</pre> <p>5. Проанализируйте, есть ли ошибка в фрагменте программы?</p> <pre>... char charr2[20] = "orange";</pre>	ОПК-2.В.1

	<pre>string str2 = "apple"; char charr3[10]; string str3; str3=str2; charr3=char2; ... 6.Проанализируйте фрагмент кода и скажите, есть ли в нем ошибка. int a=1; int b=8; const int* p=&a; p=&b;</pre>	
17	<p>1. Проанализируйте фрагмент программы и ответьте на вопрос: чему равно значение i?:</p> <pre>... int i; int ca=10; int ba=0; for (int i=0; i<=ca; i+=2) { if (ba<ca) ba+=i; else break; } std::cout<<i<<std::endl;</pre> <p>...</p> <pre> } }</pre> <p>2. Проанализируйте фрагмент программы и ответьте на вопрос: чему равно значение ds?:</p> <pre>... int f1(int); int main () { int ds,x; std::cin>>x; for(j=0;j<3;j++) ds=f1(x); std::cout << "ds="<<ds<<std::endl; }</pre>	ОПК-7.В.1

```

int fl(int y)
{
int i,as;
i=as=0;
while (i<=y)
{
as+=1;
i++;
}
return as;
}

```

Ввод: x=5;

2. Проанализируйте два фрагмента программ и определите, чему будут равны значения переменных

```

float res(float a);
int main ()
{
float x = 11,5;
cout << "res(x)="<<
res(x)<< endl;
cout << " x= " << x <<
endl;
return 0;
}
float res(float a)
{
float c=10;
a *= c;
return a;
}

```

```

float res1(float &a);
int main ()
{
float x = 11,5;
cout << "res1(x)="<<
res1(x)<< endl;
cout << " x= " << x <<
endl;
return 0;
}
float res1(float &a)
{
float c=10;
a *= c;
return a;
}

```

3. Проанализируйте, будет ли работать следующая программа? Если нет, то что нужно сделать, чтобы работала (*тип переменных не изменять!*)

```

double swapr (double & a, double & b);
int main()
{
float x=10, y=20;
swapr(x,y);
std::cout<<"x="<<x<<std::endl;
std::cout<<"y="<<y<<std::endl;
}
double swapr (double & a, double & b)
{
double temp;
temp = a+b;
return temp;
}

```

	<p>4. Проанализируйте фрагмент программы и ответьте, какой будет результат:</p> <pre>int main() { int rats = 10; int & rod = rats; int * pts = &rats; std::cout<< rod<<" "<<*pts<< std::endl; std::cout<<&rod<<" "<<pts<<std::endl; ... }</pre> <p>5. Есть фрагмент кода. Проанализируйте, есть ли в нем ошибка и, если есть, то предложите, как ее исправить.</p> <pre>enum day { Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, not_a_day }; day d; int main() { d = Sunday; while (Saturday > d) { // здесь выполняются какие-либо действия с переменной d ++d; } }</pre> <p>6. Проанализируйте, являются ли перегруженными функции с прототипами.</p> <pre>int sum(int f, int d); int sum(int& f, int& d);</pre> <p>Аргументируйте ответ</p>	
18	<p>1. Спроектируйте класс Vector3D для работы с трёхмерными векторами. Класс должен включать:</p> <ul style="list-style-type: none"> • Закрытые поля для координат (x, y, z); • Конструкторы (по умолчанию, с параметрами, копирования); • Методы для сложения, вычитания векторов и вычисления скалярного произведения; • Перегрузку операторов +, -, * (скалярное произведение), << (для вывода в поток). <p>Напишите пример использования класса.</p> <p>2. Создайте класс Matrix для работы с матрицами. Реализуйте:</p> <ul style="list-style-type: none"> • Хранение данных в динамическом массиве; • Конструкторы и деструктор; • Методы для сложения, умножения матриц, транспонирования; 	ПК-7.В.1

	<p>рования;</p> <ul style="list-style-type: none"> • Перегрузку операторов +, *, <<. • Продемонстрируйте работу класса на примере. <p>3. Реализуйте класс SmartArray — аналог std::vector, который:</p> <ul style="list-style-type: none"> • Хранит динамический массив целых чисел; • Поддерживает добавление элементов, доступ по индексу, получение размера; • Автоматически освобождает память при уничтожении объекта. <p>Объясните, как работает управление памятью в вашем классе.</p> <p>4. Спроектируйте иерархию классов для системы геометрических фигур:</p> <ul style="list-style-type: none"> • Базовый класс Shape с виртуальными методами area() и perimeter(); • Производные классы Circle, Rectangle, Triangle, переопределяющие методы area() и perimeter(); • Демонстрация полиморфизма: создайте массив указателей на Shape и выведите площадь каждой фигуры. <p>Объясните, почему методы area() и perimeter() должны быть виртуальными."</p> <p>5. Реализуйте класс Employee (сотрудник) с полями name и salary, и производные классы Manager (с полем bonus) и Developer (с полем programmingLanguage). Переопределите метод calculateAnnualSalary() для вычисления годовой зарплаты с учётом бонуса или надбавки за знание языка программирования.</p> <p>6. Реализуйте класс Animal с виртуальной функцией makeSound(), и производные классы Dog, Cat, Cow, переопределяющие эту функцию. Продемонстрируйте полиморфизм, создав массив указателей на Animal и вызвав makeSound() для каждого объекта."</p>	
--	---	--

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
	Учебным планом не предусмотрено	

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 выполнения курсовой работы
	Тема. Аппроксимация функций методом наименьших квадратов
1	Решение системы уравнений методом последовательных итераций (7 вариантов заданий)
2	Решение системы уравнений методом Зейделя (7 вариантов заданий)

3	Решение системы уравнений методом Гаусса (7 вариантов заданий)
4	Решение системы уравнений методом обратной матрицы (7 вариантов заданий)

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
	Данный вид оценочных средств не применяется при промежуточной аттестации и текущем контроле успеваемости.	

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:

- Изложение материала по рассматриваемой теме;
- Демонстрация примеров решения конкретных задач;
- Ответы на возникающие вопросы по теме лекции.

11.2. Методические указания для обучающихся по участию в семинарах

Учебным планом не предусмотрено

11.3. Методические указания для обучающихся по прохождению практических занятий

Практическое занятие является одной из основных форм организации учебного процесса, заключающаяся в выполнении обучающимися под руководством преподавателя комплекса учебных заданий с целью усвоения научно-теоретических основ учебной дисциплины, приобретения умений и навыков, опыта творческой деятельности.

Целью практического занятия для обучающегося является привитие обучающимся умений и навыков практической деятельности по изучаемой дисциплине.

Планируемые результаты при освоении обучающимся практических занятий:

- закрепление, углубление, расширение и детализация знаний при решении конкретных задач;
- развитие познавательных способностей, самостоятельности мышления, творческой активности;
- овладение новыми методами и методиками изучения конкретной учебной дисциплины;
- выработка способности логического осмысления полученных знаний для выполнения заданий;
- обеспечение рационального сочетания коллективной и индивидуальной форм обучения.

Требования к проведению практических занятий

Требования к проведению практических занятий, темы занятий и задания приведены в методических материалах, расположенных на сервере локальной сети кафедры 2.

11.4. 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.

Методические указания, задания, структура и форма отчета, а также примеры выполнения лабораторных работ изложены в методических указаниях:

3. В.А.Галанина. С.Л.Козенко, Основы программирования, ч.1/Учебно-методическое пособие – СПб, ГУАП, 2022. – 51 с.
4. В.А.Галанина.С.Л.Козенко, Основы программирования», ч..2/ Учебно-методическое пособие – СПб, ГУАП, 2023. –60с.
5. В.А.Галанина.С.Л.Козенко. Основы объектно-ориентированного программирования - СПб, ГУАП, 2025. –50 с.

11.5. Методические указания для обучающихся по прохождению курсового проектирования/выполнения курсовой работы

Курсовой проект/ работа проводится с целью формирования у обучающихся опыта комплексного решения конкретных задач профессиональной деятельности.

Курсовой проект/ работа позволяет обучающемуся:

- систематизировать и закрепить полученные теоретические знания и практические умения
- применить полученные знания, умения и практический опыт при решении практической задачи обработки результатов эксперимента
- углубить теоретические знания в соответствии с заданной темой;
- развить системное мышление, творческую инициативу, самостоятельность, организованность и ответственность за принимаемые решения;
- сформировать навыки планомерной регулярной работы над решением поставленных задач.

Методические указания, задания, структура и форма пояснительной записки, а также примеры выполнения курсовой работы изложены в методических указаниях:

- Галанина В.А., Козенко С.Л. Информатика. Методические указания к выполнению курсовой работы, ч.1.- СПб, ГУАП, 2018. – 55 с;
- Галанина В.А., Козенко С.Л. Информатика. Методические указания к выполнению курсовой работы, ч.2.- СПб, ГУАП, 2019. – 47 с;

11.6. 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;

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

В течение семестра обучающиеся загружают в ЭИОС ГУАП отчёты по лабораторным работам. Все отчеты должны быть загружены до начала зачетной недели.

Текущий контроль осуществляется в течение семестра в процессе защиты лабораторных работ и выставления соответствующих баллов за работу. Для каждой лабораторной работы указывается предельный срок ее сдачи и максимально возможное количество баллов (10 баллов). В случае сдачи работы после указанного срока, снижается максимально возможное количество баллов за данную работу (за неделю пропуска срока снимается 1 балл). В конце семестра подсчитывается общее количество баллов за лабораторные работы и выставляется оценка в соответствии с таблицей:

Баллы за ЛР	Оценка
54-60	<i>Отлично</i>
48-53	<i>Хорошо</i>
31-47	<i>Удовл</i>
≤ 30	<i>Недопуск</i>

Текущий контроль за выполнением курсового проектирования осуществляется трижды в семестре по результатам выполнения промежуточных этапов выполнения курсовой работы. Каждый этап оценивается в баллах (20 баллов за этап). В случае успешного прохождения промежуточной аттестации в виде защиты курсовой работы результаты прохождения текущего контроля учитываются следующим образом:

-получение аттестационной оценки «*отлично*» за курсовую работу возможно при получении за промежуточные этапы не менее 85% баллов от максимально возможного, оценки «*хорошо*» - при получении не менее 75% баллов от максимально возможного, оценки «*удовлетворительно*» - при получении не менее 55% баллов от максимально возможного.

11.8. 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 – форму оценки знаний, полученных обучающимися в процессе изучения всей дисциплины и ее части, навыков самостоятельной работы, способности применять их для решения практических задач. Экзамен проводится в устной форме в период экзаменационной сессии и завершается аттестационной оценкой «*отлично*», «*хорошо*», «*удовлетворительно*», «*неудовлетворительно*».

Для допуска к экзамену должны быть сданы все лабораторные работы и получена итоговая положительная оценка.

Оценка за экзамен ставится по пятибалльной системе в соответствии с таблицей 14.

Аттестационная оценка определяется, как средняя между оценкой за лабораторные работы и оценкой за экзамен. При разнице оценок в 1 балл берется оценка за экзамен.

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