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

Кафедра № 42

УТВЕРЖДАЮ
Руководитель образовательной программы

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

В.А. Миклуш
(инициалы, фамилия)

(подпись)
«29» января 2026 г

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

«Основы программирования»
(Наименование дисциплины)

Код направления подготовки/ специальности	09.03.02
Наименование направления подготовки/ специальности	Информационные системы и технологии
Наименование направленности/ специализации	Информационные технологии в медиаиндустрии
Форма обучения	очная
Год приема	2026

Санкт-Петербург– 2026

Лист согласования рабочей программы дисциплины

Программу составил (а)

доц., к.т.н., доц..	29.01.2026	Н.В. Богословская
доц., к.т.н., доц	29.01.2026	В.А. Кузнецов
(должность, уч. степень, звание)	(подпись, дата)	(инициалы, фамилия)

Программа одобрена на заседании кафедры № 42
29. 01 2026 г., протокол № 05/2025-26

Заведующий кафедрой № 42

д.т.н., доц.	29.01.2026	С.В. Мичурин
(уч. степень, звание)	(подпись, дата)	(инициалы, фамилия)

Заместитель директора института №4 по методической работе

доц., к.т.н.	29.01.2026	А.А. Фоменкова
(должность, уч. степень, звание)	(подпись, дата)	(инициалы, фамилия)

Аннотация

Дисциплина «Основы программирования» входит в образовательную программу высшего образования – программу бакалавриата по направлению подготовки/ специальности 09.03.02 «Информационные системы и технологии» направленности/специализации «Информационные технологии в медиаиндустрии». Дисциплина реализуется кафедрой «№42».

Дисциплина нацелена на формирование у выпускника следующих компетенций:

УК-1 «Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач»

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-1 «Способен применять естественнонаучные и общеинженерные знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности»

ОПК-6 «Способен разрабатывать алгоритмы и программы, пригодные для практического применения в области информационных систем и технологий»

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 (ОПК-1).

В практической части дисциплины формируются навыки объектного программирования (ОПК-6), 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.

В процессе освоения дисциплины продолжается формирование универсальных компетенций УК-1 в части системного подхода и УК-2 в выборе оптимальных способов решения поставленных задач с учетом имеющихся ресурсов и ограничений.

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 часов.

Язык обучения по дисциплине «русский».

1. Перечень планируемых результатов обучения по дисциплине

1.1. Цели преподавания дисциплины - формирование у студентов базовых знаний и навыков в области объектно-ориентированного программирования как фундаментальной основы для профессиональной деятельности в сфере информационных технологий. В рамках курса студенты должны освоить ключевые концепции объектно-ориентированного подхода, научиться применять их для теоретического и экспериментального исследования, а также получить практический опыт использования современных инструментов объектно-ориентированной разработки для решения прикладных задач.

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-1 Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	УК-1.У.1 уметь применять методики поиска, сбора и обработки информации, в том числе с использованием искусственного интеллекта
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.З.3 знать возможности и ограничения применения цифровых инструментов для решения поставленных задач УК-2.У.1 уметь проводить анализ поставленной цели и формулировать задачи, которые необходимо решить для ее достижения УК-2.У.3 уметь выдвигать альтернативные варианты действий с целью выбора оптимальных способов решения задач, в том числе с помощью цифровых средств
Общепрофессиональные компетенции	ОПК-1 Способен применять естественнонаучные и общеинженерные знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в	ОПК-1.З.1 знать основы математики, физики, вычислительной техники и программирования ОПК-1.У.1 уметь решать стандартные профессиональные задачи с применением естественнонаучных и общеинженерных знаний, методов математического анализа и моделирования ОПК-1.В.1 иметь навыки теоретического и экспериментального исследования

	профессиональной деятельности	объектов профессиональной деятельности
Общепрофессиональные компетенции	ОПК-6 Способен разрабатывать алгоритмы и программы, пригодные для практического применения в области информационных систем и технологий	ОПК-6.З.1 знать методы алгоритмизации, языки и технологии программирования, пригодные для практического применения в области информационных систем и технологий ОПК-6.У.1 уметь применять методы алгоритмизации, языки и технологии программирования при решении профессиональных задач в области информационных систем и технологий ОПК-6.В.1 иметь навыки программирования, отладки и тестирования прототипов программно-технических комплексов задач

2. Место дисциплины в структуре ОП

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

- Дискретная математика,
- Информатика,
- Основы цифровой грамотности.

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

- Алгоритмы и структуры данных,
- Технологии программирования,
- Кроссплатформенное программирование,
- Разработка и анализ требований,
- Большие данные.

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			
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.

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/КР (час)	СР (час)
Семестр 2					
Раздел 1. Основы языка С/С++ Тема 1.1. Типы данных. Преобразование типов Тема 1.2. Структура программы. Библиотеки. Компилятор Тема 1.3 Синтаксис и операторы языка С	6	6	8		20
Раздел 2. Инструменты языка С/С++ 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	20	20	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	8	8	8		20
Итого в семестре:	34	34	34		60
Семестр 3					
Раздел 4. Архитектура классов	24		18		15
Раздел 5. Библиотеки классов в разработке приложений пользователя	10		16		15
Выполнение курсовой работы				17	29
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	Раздел 1. Основы языка C/C++ 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. Тема 1.2. Структура программы. Библиотеки. Компилятор Подключение библиотек. Заголовки. Стандартные библиотеки. Функция main(). Глобальные и локальные переменные. Компиляция. Запуск и отладка программы. 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.
2	Раздел 2. Инструменты языка C/C++ 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. Тема 2.2. Указатель, ссылки Косвенная адресация. Переменные и указатели на них. Получение адреса. Операторы * и &. 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. Тема 2.4. Символы, строки, хеширование Основные таблицы символов ASCII, CP-1251, UTF-8, UTF-16. Обработка строк с использованием <stdio>, <iostream>. Хеширование строк. Тема 2.5. Рекурсивные функции Понятие рекурсии и рекурсивной функции. Условие завершения рекурсивного вызова, бесконечный вызов рекурсивной функции. Косвенная рекурсия. Глубина рекурсии. Тема 2.6 Структуры Объявление структур в языке C. Доступ к элементам структуры. Выравнивание структур. Тема 2.7. Поразрядные операции
	Поразрядные или побитовые операторы “и”&,“или” |,“не” ~, “исключающее или” ^, операторы сдвига >>,<<. Циклический сдвиг.
3	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. Оценка времени доступа к данным и поиска для структур данных: массив, список, дерево. Тема 3.3. Сортировка пузырьком. Сортировка расческой. Гномья сортировка Тема 3.4. Сортировка вставками. Сортировка Шелла
4	Раздел 4. Архитектура классов Тема 4.1 Базовые принципы объектно-ориентированного программирования: Инкапсуляция. Классы и объекты. Модификаторы доступа. Поля классов. Разработка библиотеки классов. Тема 4.2 Наследование классов. Реализация наследования: перегрузка и переопределение. Назначение конструкторов класса. Статические конструкторы. Тема 4.3. Свойства классов – механизм доступа к данным класса. Виртуальные свойства. Тема 4.4 Полиморфизм. Методы классов: перегрузка и переопределение. Виртуальные методы. Сокрытие методов. Параметры методов. Тема 4.5 Иерархия классов. Абстрактные классы. Обобщенные типы. Класс System.Object. Назначение абстрактных классов. Наследование обобщенных типов. Тема 4.6 События классов. Обработчики событий. Технология создания события и подписки на событие. Практические примеры использования событий классов. Тема 4.7 Интерфейсы – механизм множественного наследования в C#. Назначение интерфейсов. Технология разработки интерфейсов и подписки классов. Практические примеры использования интерфейсов для реализации множественного наследования. Тема 4.8 Отношения между классами и их реализация. Диаграмма классов в UML. Реализация отношений в структуре классов. Практические приемы использования отношений классов. Универсальные шаблоны
5	Раздел 5. Библиотеки классов в разработке приложений пользователя Тема 5.1 Технологии доступа к удаленным данным в информационных системах. Классы и таблицы баз данных. Установление соединения с сервером баз данных из

	приложений пользователя. Тема 5.2 Библиотека ADO .NET. Класс Connection. Свойства и методы класса Command. Тема 5.3 Разработка интерфейса пользователя средствами классов библиотеки ADO .NET

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
Семестр 2					
	Знакомство со средой разработки и языком С. Функции	Решение задач	8		1
	Указатели, Массивы и Циклы	Решение задач	8		2
	Простые рекурсивные функции	Решение задач	10		2
	Поразрядные операции	Решение задач	8		2
Всего			34		

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	Алгоритм с использование условного оператора	2		1
2	Алгоритм с использованием циклов	2		1
3	Одномерные массивы	4		1
4	Посимвольная обработка строк	2		2
5	Анализ строк и хеширование	4		2
6	Рекурсивный алгоритм	2		2
7	Алгоритм с использование рекурсии и циклов	4		2
8	Бинарные (битовые) операции	2		2
9	Обработка битовых блоков и последовательностей	4		2
10	Способы измерения времени работы алгоритмов	4		3
11	Оценка сложности алгоритмов	4		3
Семестр 3				
1	Классы и объекты. Поля и конструктор по умолчанию. Проект библиотеки	2		4
2	Свойства класса. Методы класса	4		4
3	Параметры методов	4		4
4	Переопределение свойств и методов	2		4
5	Отношения между классами	4		4
6	События классов	2		4
7	Классы и таблицы баз данных	4		5
8	Библиотека классов доступа к данным	4		5
9	Представление данных в приложении	4		5
10	Интерфейс приложения базы данных	4		5
Всего		68		

4.5. Выполнение курсового проекта/ курсовой работы

Цель курсовой работы:

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

Цель курсовой работы: использование полученных знаний и навыков объектно-ориентированного программирования для создания пользовательской библиотеки классов, предназначенной для разработки приложений пользователя в конкретной предметной области.

Библиотека должна учитывать особенности области приложения и располагать необходимой функциональностью для построения интерфейса пользователя, структурирования, записи, чтения данных в различных форматах, обработки, вычисления и визуализации данных.

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)		20	15
Курсовое проектирование (КП, КР)			15
Расчетно-графические задания (РГЗ)			
Выполнение реферата (Р)			
Подготовка к текущему контролю успеваемости (ТКУ)		20	9
Домашнее задание (ДЗ)			
Контрольные работы заочников (КРЗ)			
Подготовка к промежуточной аттестации (ПА)		20	20
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. разделов 6-11.

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)
Библиотека 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	
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.	
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	
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»

URL адрес	Наименование
https://docs.microsoft.com/ru-ru/dotnet/csharp/	Docs / .NET / Руководство по языку C#

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

Примечание: *экзаменационные билеты формируются на основе вопросов и задач таблицы 15.

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»	Обучающийся: – глубоко и всесторонне усвоил программный материал; – уверенно, логично, последовательно и грамотно его излагает; – опираясь на знания основной и дополнительной литературы, тесно связывает усвоенные научные положения с практической деятельностью направления; – умело обосновывает и аргументирует выдвигаемые им идеи; – делает выводы и обобщения; – свободно владеет системой специализированных понятий. – правильно выполнил от 90% до 100% тестовых заданий.
«хорошо» «зачтено»	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. – правильно выполнил от 70% до 89% тестовых заданий.
«удовлетворительно» «зачтено»	– 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. – правильно выполнил от 51% до 69% тестовых заданий.
«неудовлетворительно» «не зачтено»	– 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. – правильно выполнил менее 51% тестовых заданий.

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
	Семестр 2	
1	Структура программного кода, компиляция, файл кода программы.	ОПК-1.3.1
2	Библиотеки. Заголовки. Подключение библиотек и заголовочных файлов.	ОПК-1.3.1
3	Синтаксис С++. Операторы языка. Бинарные, унарные операторы. Префиксные, постфиксные.	ОПК-1.3.1
4	Условный оператор. Особенности значений выражения условного оператора.	ОПК-1.3.1

5	Циклы, операторы прерывания.	ОПК-1.З.1
6	Переменные. Типы данных: статическая и динамическая типизация в разных языках программирования.	ОПК-1.З.1
7	Типы данных. Преобразование типов, явное, неявное.	ОПК-1.З.1
8	Типы данных. Пользовательские типы данных. Массивы.	ОПК-1.З.1
9	Доступ к элементу массива, поиск элемента массива. Бинарный поиск в отсортированном массиве.	ОПК-6.У.1
10	Указатели, ссылки. Принципы доступа к данным.	ОПК-6.З.1
11	Ссылки и указатели в качестве аргумента функции.	ОПК-6.З.1
12	Пустой указатель, преобразование указателя.	ОПК-1.З.1
13	Динамическое выделение памяти.	ОПК-1.З.1
14	Выделение памяти для многомерного массива.	ОПК-6.З.1
15	Представление области памяти в виде двумерных и трехмерных массивов.	ОПК-6.З.1
16	Структуры. Динамические структуры. Стеки, списки, деревья.	ОПК-1.З.1
17	Двоичное представление данных. Бинарные операции.	ОПК-1.З.1
18	Простейшие алгоритмы с использованием бинарных операций.	ОПК-6.У.1
19	Побитовые операции над типами с плавающей запятой.	ОПК-1.З.1
20	Символы. Таблицы кодировок символов.	ОПК-1.З.1
21	Строки. Представление строковых переменных.	ОПК-1.В.1
22	Хэширование.	УК-1.У.1
23	Потоковый ввод/вывод.	ОПК-1.З.1
24	Форматированный ввод/вывод. Вывод на консоль и в файл.	УК-2.З.3
25	Функции. Прототип функции. Параметр по умолчанию.	ОПК-1.З.1
26	Перегрузка функций.	ОПК-1.З.1
27	Рекурсия. Рекурсивные функции.	ОПК-6.З.1
28	Встраиваемые функции. Макросы.	ОПК-1.З.1
29	Асимптотическая оценка сложности алгоритмов.	ОПК-6.З.1
30	Оценка сложности алгоритма. Оценка сложности циклов, процедуры поиска и доступа к элементу.	ОПК-1.В.1
31	Оценка сложности алгоритма. Измерение времени работы алгоритма.	ОПК-1.В.1
32	Задача сортировки массива. Сортировки: гномья, вставками, пузырьком, Шелла, расческой.	УК-1.У.1

33	Чему равно значение “*c” после выполнения команд?
	`int b[8] = { 7,2 ,11,1,2,11,8,0 };`
	`int* c;`
	`for (c = b; *c < 10; c++)`
	`*(c + 1) = *c + *(c + 1);`
		ОПК-6.В.1

34	Что выведет на экран данный код?
	`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()`
		ОПК-6.В.1

	`{`
	`if (Compare(3, 7) || Compare(7, 3))`
	`printf("inevitability; ");`	

35	Сколько аргументов может быть при вызове функции int F(float A[4], float B[4], float C)?	ОПК-6.В.1

36	Что возвращает функция
	`int F(int A, int B)`
	`{`
	`if (A > B)`
	`if (B > 100)`
	`B = 100;`
	`else`
	`B = A;`
	`return A + B;`
	`}`
	`при вызове F(230, 90);`
		ОПК-6.У.1

37	Какой код выводит на экран все нечетные числа от 19 до 3, каждое число с новой строки?	ОПК-6.У.1
	3 семестр	
1	Классы и структуры как основные конструкции системы общих типов CTS, используемой на платформе .NET.	УК-1.У.1
2	Общая система типов платформы .NET (Common Type System – CTS). Основные конструкции CTS. Назначение класса Object.	УК-2.З.3
3	Иерархия классов. Реализация наследования классов. Доступ к членам базового класса из класса-наследника.	УК-2.У.1
4	Преобразование типов в иерархии классов. Сокрытие свойств и методов класса.	УК-2.У.3
5	Методы класса. Передача параметров методу. Статические методы и методы экземпляра.	ОПК-1.З.1
6	Унаследованные и переопределенные методы. Перегрузка методов – частный случай полиморфизма. Скрытие производного класса членов базового класса.	ОПК-1.У.1
7	Назначение и реализация интерфейса. Работа с объектами через интерфейсы. Интерфейсы и множественное наследование. Интерфейсы в преобразованиях типов.	ОПК-1.В.1
8	Классы и объекты. Поля и конструктор по умолчанию. Инициализаторы объектов класса.	ОПК-6.З.1
9	Уровни доступности типов и членов типов.	ОПК-6.У.1
10	Конструкторы экземпляров и статические конструкторы. Статические классы.	ОПК-6.В.1
11	Данные классов: поля как переменные класса и константные значения, связанные с классом. Модификаторы static, const, readonly.	ОПК-6.З.1
12	Свойства класса. Свойства экземпляра, статические свойства. Автоматически реализуемые свойства. Переопределение свойств класса.	ОПК-6.З.1
13	Унаследованные и переопределенные методы. Перегрузка методов – частный случай полиморфизма. Скрытие производного класса членов базового класса.	ОПК-6.З.1
14	Преобразование типов в иерархии классов. Сокрытие	ОПК-6.З.1

	свойств и методов класса.	
15	Основные отношения между классами: наследование, реализация. Одиночное и множественное наследование. Неявное наследование.	УК-1.У.1
16	Основные отношения между классами: ассоциация, композиция и агрегация.	УК-1.У.1
17	События классов. Определение и вызов событий. Класс данных события EventArgs.	ОПК-6.У.1

18	Имеется описание класса Rectangle
	public class Rectangle {
	double length;
	double width;
	public Rectangle (double l, double w) {
	length = l;
	width = w;
	}
	public double getArea() {
	area = length*width;
	return (area);
	}
	}
	Обеспечьте следующие условия: класс Circle также имеет метод getArea, изначально определенный как abstract в классе Shape. Напишите программный код классов.
		ОПК-6.У.1

19	Объекты классов Circle и Rectangle могут быть созданы следующими строками кода: Circle circle = new Circle(5); Rectangle rectangle = new Rectangle(4,5); Напишите код классов Circle и Rectangle.	ОПК-6.У.1

20	Имеется описание класса Shape:
	public abstract class Shape
	{
	protected double area;
	public abstract double calcArea();
	}
	Предложите вариант развития иерархии классов.
		ОПК-6.У.1

21	Имеется фрагмент описания класса
	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;}
	} Создайте объекты класса всеми возможными способами.
		ОПК-6.У.1

22	Используя отношение композиции опишите структуру классов для следующей фразы «Руль – часть автомобиля».	ОПК-6.З.1
23	Используя отношение агрегации опишите структуру классов для следующей фразы: «Компьютерная система — это целое. Компонентами являются монитор, клавиатура, мышь и системный блок компьютера».	ОПК-6.З.1
24	Опишите структуру классов Employee и Child, с учетом того, что у сотрудника может быть любое количество детей.	ОПК-6.У.1
25	Опишите структуру классов Employee и Department, с учетом того, что при определенных обстоятельствах работник может быть ассоциирован с несколькими отделами.	ОПК-6.У.1

26	Разработайте структуру класса Employee, для которого можно выполнить следующие действия:
	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;
		ОПК-6.В.1

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

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 выполнения курсового проекта/ курсовой работы представлены в таблице 17.

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

№ п/п	Примерный перечень тем для выполнения курсовой работы / выполнения курсового проекта
1	В приложении пользователь может создать объект класса Текст, используя классы Предложение, Слово. Классы должны иметь методы для создания, изменения, чтения из файла, сохранения в файл, удаления.
2	В приложении пользователь может создать объект класса Автомобиль, используя классы Колесо, Двигатель. Методы классов должны фиксировать следующие состояния: движение, заправка, замена колеса. Информация об автомобиле должна включать все технические особенности.
3	В приложении пользователь может создать объект класса Самолет, используя классы Крыло, Шасси, Двигатель. Реализовать методы для сборки самолета, описания маршрута, заправки.
4	В приложении пользователь может создать объект класса Государство, используя классы Область, Район, Город. Реализовать методы чтения, записи необходимой информации о государстве, обеспечить поиск областей, районов, городов с соответствующей информацией о каждом объекте.
5	В приложении пользователь может создать объект класса Планета, используя классы Материк, Океан, Остров. Методы для чтения, записи всей необходимой

	информации о планете, поиск и сортировку всех объектов, изменение информации.
6	В приложении пользователь может создать объект класса Звездная система, используя классы Планета, Звезда, Луна. Необходимы методы чтения, записи, просмотра списков объектов, изменения информации.
7	В приложении пользователь может создать объект класса Компьютер, используя классы Винчестер, Дисковод, Оперативная память, Процессор. Методы для проверки на вирусы, сборки, чтения, записи полной технической информации о собранном компьютере.
8	В приложении пользователь может создать объект класса Квадрат, используя классы Точка, Отрезок. Необходимы методы для задания размеров, цвета заливки и прорисовки контуров фигур, создания, удаления, перемещения, изменения цвета.
9	В приложении пользователь может создать объект класса Круг, используя классы Точка, Окружность. Методы для работы с фигурами должны обеспечить создание, удаление, изменение размеров и координат, определения принадлежности точки, перемещения.
10	В приложении пользователь может создать объект класса Щенок, используя классы Животное, Собака. Методы для работы с информацией о щенке должны позволять читать, сохранять всю информацию о щенке, обеспечить просмотр фотографий щенка с возможностью чтения, сохранения.
11	В приложении пользователь может создать объект класса Текстовый файл, используя классы Файл, Директория. Методы для работы с файлами должны полностью обеспечить пользователя возможностями создания, удаления, изменения, переименования файлов.
12	В приложении пользователь может создать объект класса Одномерный массив, используя классы Массив, Элемент. Методы для работы с массивами должны обеспечивать их создание, изменение, выполнение операций (сложить, вычесть, перемножить).
13	В приложении пользователь может создать объект класса Простая дробь, используя класс Число. Методы для работы с объектами должны обеспечить создание, выполнение операций сложения, вычитания, умножения и деления.
14	В приложении пользователь может создать объект класса Дом, используя классы Окно, Дверь. Методы должны позволять сохранить, прочитать, изменить всю необходимую информацию об объекте, показать графическое представление объекта с возможностью сохранения и редактирования.
15	В приложении пользователь может создать объект класса Цветок, используя классы Лепесток, Бутон. Методы для работы с объектами должны сохранять, изменять всю необходимую информацию как в текстовом, так и в графическом виде.
16	В приложении пользователь может создать объект класса Дерево, используя классы Лист, Ветка. Методы для описания объектов должны позволять сохранять, изменять всю необходимую информацию об объекте: посадке, обрезке, прививке как в текстовом, так и в графическом виде.
17	Создать базовый класс Point. В приложении пользователь может на его основе создать классы ColoredPoint и Line. На основе класса Line создать классы ColoredLine и PolyLine. Методы для работы с графическими объектами должны позволять создание, заливку, удаление, изменение цвета контура, перемещение.
18	В приложении пользователь может создать объект класса Фотоальбом, используя классы Фотография, Страница. Методы для работы с графическими объектами должны обеспечить создание, удаление, редактирование и просмотр.
19	В приложении пользователь может описать класс Комната, содержащий сведения о метраже, высоте потолков и количестве окон. Описать методы для создания, изменения, вычисления параметров (например, площади), записи, поиска и сортировки в каталоге.
20	В приложении пользователь может создать класс Поезд, содержащий следующую информацию: пункт назначения, номер поезда (может содержать буквы и цифры, время и дату отправления, количество вагонов различного типа (плацкартных, общих, купе, СВ). Предусмотреть свойства для получения состояния объекта. Описать класс Вокзал, содержащий массив поездов. Обеспечить следующие возможности: вывод информации о поезде по указанному номеру, вывод информации о поездах, отправляющихся после указанного времени, или в указанном направлении, или по совокупности параметров времени и направления.
21	В приложении пользователь может создать объект класса Самолет, содержащий следующую информацию: пункта назначения; номер рейса; название авиакомпании; дату и время отправления; стоимость полета. Предусмотреть свойства для получения состояния объекта. Описать класс Аэропорт, содержащий массив самолетов. Обеспечить следующие возможности: вывод информации о самолете по указанному номеру рейса; вывод информации о самолетах, отправляющихся после указанного времени, или в указанном направлении, или по совокупности параметров времени и направления.
22	В приложении пользователь может создать объект класса Товар, содержащий следующие закрытые поля: название товара; название фирмы-изготовителя; стоимость товара. Предусмотреть свойства для получения состояния объекта. Описать класс Склад, содержащий массив товаров. Обеспечить следующие возможности: вывод информации о товаре по введенному наименованию; поиск подходящего товара по наименованию и цене.
23	В приложении пользователь может создать объект класса Туристическая путевка. Сформировать набор предложений клиенту по выбору туристической путевки различного типа (отдых, экскурсии, лечение, шопинг, круиз и т. д.) для оптимального выбора. Учитывать возможность выбора транспорта, питания и числа дней. Реализовать выбор и сортировку направлений.

24	Создать абстрактный класс Figura. В приложении пользователь может на его основе создать классы Triangle и Square. На основе класса Square создать класс ManySided. Методы для работы с фигурами должны обеспечивать установку и получение значений всех координат, а также для изменения цвета и получения текущего цвета; для треугольника – метод, вычисляющий возможность построения треугольника; для всех фигур – методы масштабирования; для квадрата и многоугольника – методы заливки внутреннего пространства; для всех фигур – методы вывода на заданную пользователем поверхность и методы удаления (уничтожения, стирания) фигур с поверхности. Класс Triangle объявить закрытым классом, не имеющим наследников.
25	В приложении пользователь может создать класс Phone, который содержит описание его технических параметров и реализует методы для ответа на звонок, совершения звонка, приема и отправки сообщения, работы с фото и видео и т.д. Определить иерархию Телефоны, Мобильные телефоны, Проводные телефоны, Смартфоны, Кнопочные телефоны, Фитнес-часы, Фитнес-браслеты.
26	В приложении пользователь может создать объект класса Транспорт. Определить иерархию подвижного состава железнодорожного транспорта. Создать пассажирский поезд. Подсчитать общую численность пассажиров и багажа. Провести сортировку вагонов поезда на основе уровня комфортности. Найти в поезде вагоны, соответствующие заданному диапазону параметров числа пассажиров.
27	В приложении пользователь может создать объект класса Авиакомпания. Определить иерархию самолетов. Провести сортировку самолетов компании по дальности полета. Найти самолет в компании, соответствующий заданному диапазону параметров потребления горючего.
28	В приложении пользователь может создать объект класса Таксопарк. Определить иерархию легковых автомобилей. Провести сортировку автомобилей парка по расходу топлива. Найти автомобиль в компании, соответствующий заданному диапазону параметров скорости.
29	В приложении пользователь может создать объект класса Кредит. Сформировать набор предложений клиенту по целевым кредитам различных банков для оптимального выбора. Учитывать возможность досрочного погашения кредита и/или увеличения кредитной линии. Реализовать выбор и поиск кредита.
30	В приложении пользователь может создать объект класса Счет и выполнить все необходимые действия. Необходимо учитывать следующие возможности: клиент может иметь несколько счетов в банке; счет может быть открыт, закрыт; счет открыт с определенными условиями (ставка, срок). Требуется обеспечить начисление и списывание сумм по счетам, начисление процентов в соответствие с условиями счета (процентная ставка и сроки начисления), списывание оплаты за ведение счета. Реализовать необходимые отношения с классами Клиент, Банк,
31	<Тема самостоятельно сформулирована студентом и утверждена у руководителя>

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	*Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа.* Чем отличается объектно-ориентированный стиль программирования от структурного? 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 достижения поставленной цели. 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. 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. 4 Структурное программирование основано на модели построения программы как иерархии процедур. Для ООП основным принципом является отсутствие явного описания последовательности действий для выполнения программы.	УК-1
2	*Прочитайте текст, выберите правильные варианты ответа и запишите аргументы, обосновывающие выбор ответов.* Ключевыми принципами объектно-ориентированного программирования являются: 1 Абстракция. Моделирование требуемых атрибутов и взаимодействий сущностей в виде классов для определения абстрактного представления системы. 2 Инкапсуляция. Скрытие внутреннего состояния и функций объекта и предоставление доступа только через открытый набор функций. 3 Наследование. Возможность создания новых абстракций на основе существующих. 4 Полиморфизм. Возможность реализации наследуемых свойств или методов отличающимися способами в рамках множества абстракций. 5 Формализация. Отображение результатов мышления в точных понятиях и утверждениях.	УК-1
3	*Прочитайте текст и установите соответствие. К каждой позиции, данной в левом столбце, подберите соответствующую позицию в правом столбце.* 1. Система программирования — это А. комплекс программных средств, предназначенных для поддержки программного продукта на протяжении всего жизненного цикла этого	УК-1

	продукта	
2. Система — это	B. 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	*Прочитайте текст и установите последовательность. Запишите соответствующую последовательность букв слева направо.* В зависимости от вида, масштабов и потребностей программного проекта определяется порядок разработки: A. Разработка и анализ требований B. Проектирование C. Дизайн D. Кодирование E. Тестирование F. Внедрение G. Сопровождение	УК-1
5	*Прочитайте текст и запишите развернутый обоснованный ответ.* Какие парадигмы и стили программирования поддерживает C#	УК-1
6	*Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа.* Изменение программного обеспечения (ПО): 1. Любое изменение ПО является модификацией. 2. Любое изменение ПО является модификацией, кроме случаев адаптации. 3. Изменения, которые необходимо внести в ПО, связаны с исправлением ошибок в системе. 4. Изменение ПО должно быть выполнено в специализированных приложениях.	УК-2
7	*Прочитайте текст, выберите правильные варианты ответа и запишите аргументы, обосновывающие выбор ответов.* Программное обеспечение (ПО) можно найти в этих реестрах: 1. Реестр российского программного обеспечения (Минцифры); 2. Реестр программ для ЭВМ (Роспатент, ФИПС); 3. Реестр баз данных (Роспатент, ФИПС). 4. Реестр федерального программного обеспечения (Минцифры).	УК-2
8	*Прочитайте текст и установите соответствие. К каждой позиции, данной в левом столбце, подберите соответствующую позицию в правом столбце.*	УК-2

1. Разработка ПО должна производиться без стороннего ПО, если на его использование нет разрешения правообладателя.	A. Разрешение можно получить, заключив лицензионный договор.
2. Договор заключают как обычным способом – в простой письменной форме, так и в форме договора присоединения (ст. 428 ГК РФ).	B. Текст договора можно попросить у правообладателя (вендора) или дистрибьютора (если заключили сублицензионный договор), а также найти на сайте правообладателя или в составе документации к ПО, в том числе при его установке.
3. Некоторые открытые лицензии дают возможность использовать ПО без ограничений в своих разработках, но далеко не все.	C. С содержанием ряда лицензий можно ознакомиться на сайте Open Source Initiative.

9	*Прочитайте текст и установите последовательность. Запишите соответствующую последовательность букв слева направо.* Разработка ПО в России выполняется на основании договора услуг. Важными позициями, которые должны быть прописаны в договоре в определенном порядке, являются: A. Описание работ и сроки выполнения. B. Уточнение ответственности сторон. C. Права на интеллектуальную собственность. D. Расширение обязательств и возможность изменения условий.	УК-2
10	*Прочитайте текст и запишите развернутый обоснованный ответ.* Одним из основных способов защиты прав разработчика ПО является заключение договора услуг. Какие позиции взаимодействия разработчика ПО и заказчика должны быть отражены в договоре.	УК-2
11	*Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа.* 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	ОПК-1
12	*Прочитайте текст, выберите правильные варианты ответа и запишите аргументы, обосновывающие выбор ответов.* 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 дарвинской эволюции, где генетические алгоритмы применяются для генерации новых вариантов кода и отбора лучших. 4. Экспериментальное моделирование.	ОПК-1

<table>
<tr><td>13</td><td>Прочитайте текст и установите соответствие. К каждой позиции, данной в левом столбце, подберите соответствующую позицию в правом столбце.<table><tr><td>1. Методы линейного программирования позволяют</td><td>A. 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>B. нахождения локальных оптимумов и оптимизации функций производительности кода</td></tr><tr><td>3. Эволюционное моделирование основано на принципах</td><td>C. естественного отбора и дарвинской эволюции, где генетические алгоритмы применяются для генерации новых вариантов кода и отбора лучших.</td></tr></table></td><td>ОПК-1</td></tr>
<tr><td>14</td><td>Прочитайте текст и установите последовательность. Запишите соответствующую последовательность букв слева направо.
Несмотря на появление новых тенденций, основные этапы разработки ПО остались неизменными:
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.</td><td>ОПК-1</td></tr>
<tr><td>15</td><td>Прочитайте текст и запишите развернутый обоснованный ответ.
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.</td><td>ОПК-1</td></tr>
<tr><td>16</td><td>Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа.
Что означает O(N) для алгоритма?
1. Алгоритм не имеет асимптотическое ограничение сверху
2. Алгоритм имеет линейный рост сложности
3. Время работы алгоритма зависит только от параметра N и не от чего больше
4. Алгоритм имеет постоянную сложность равную некоторой константе</td><td>ОПК-6</td></tr>
<tr><td>17</td><td>Прочитайте текст, выберите правильные варианты ответа и запишите аргументы, обосновывающие выбор ответов.
Алгоритм бинарного поиска числа в структуре размерности N соответствует следующим утверждениям:
1. Может быть реализован при помощи рекурсивной функции
2. При применении к неотсортированному массиву выполняется за O(N)
3. Является примером жадного алгоритма
4. При работе с однонаправленным списком не выполняется гарантированно за O(logN)
5. Если искомое число является максимумом, то позволяет найти его в</td><td>ОПК-6</td></tr>
<tr><td></td><td>массиве за O(1)</td><td></td></tr>
<tr><td>18</td><td>Прочитайте текст и установите соответствие. К каждой позиции, данной в левом столбце, подберите соответствующую позицию в правом столбце.<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 среднего случая</td></tr></table></td><td>ОПК-6</td></tr>
<tr><td>19</td><td>Прочитайте текст и установите последовательность. Запишите соответствующую последовательность букв слева направо.
Приведите классы сложности в порядке убывания скорости роста (сначала тот, что обладает наименьшей скорость роста при N -> бесконечность)
A. $(N^2 \log N)$
B. $(N \log^2 N)$
C. $(N^2 \log^2 N)$
D. $(N \log^3 N)$</td><td>ОПК-6</td></tr>
<tr><td>20</td><td>Прочитайте текст и запишите развернутый обоснованный ответ.
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?</td><td>ОПК-6</td></tr>
<tr><td>21</td><td>Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа.
Сколько раз выполняется преобразование типов в следующем коде C++?
int A = 10;
float B = (int) A / 10;
bool C = A > 10;
if (A)
B = C;
1. 2
2. 3
3. 4
4. 5
5. 0</td><td>ОПК-6</td></tr>
<tr><td>21</td><td>Прочитайте текст, выберите правильные варианты ответа и запишите аргументы, обосновывающие выбор ответов.
Потоковый <iostream> ввод/вывод отличается от форматированного <stdio.h> ввода/вывода C++ тем, что:
1. Потоковый ввод/вывод не позволяет форматировать данные для вывода
2. Наименованием библиотеки, в которой реализованы методы</td><td>ОПК-6</td></tr>
</table>

<table>
<tr><td></td><td>3. Различным механизмом работы с классами
4. Форматированный ввод/вывод требует указания типа данных аргументов
5. Форматированный вывод <stdio.h> не позволяет выводить данные в буферный массив</td><td></td></tr>
<tr><td>23</td><td>Прочитайте текст и установите соответствие. К каждой позиции, данной в левом столбце, подберите соответствующую позицию в правом столбце.

Приведите в соответствие размеры структур языка C++, в байтах, 32 разрядная система

<table>
<tr><td>1. struct DataStr1
{
float N;
float Data[11];
short* ptr;
};</td><td>A. 56</td></tr>
<tr><td>2. struct DataStr2
{
float Score;
char Name[20];
char Surname[30];

};</td><td>B. 52</td></tr>
<tr><td>3. struct DataStr3
{
double[6] Values;
float Mean;
DataStr3* next;
};</td><td>C. 54</td></tr>
</table>
</td><td>ОПК-6</td></tr>
<tr><td>24</td><td>Прочитайте текст и установите последовательность. Запишите соответствующую последовательность букв слева направо.

Расположите следующие структуры данных по скорости доступа к элементу по его номеру в структуре:
A. Массив
B. Однонаправленный список
C. Дерево (хранение в массиве последовательно по уровням)
D. Двунаправленный список</td><td>ОПК-6</td></tr>
</table>

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 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

Проведение практических занятий требует от студента обязательной работы с конспектом лекций. В конце занятия предусмотрено подведение итогов: указываются недостатки и достигнутые успехи, ответы на вопросы студентов, пути дальнейшего совершенствования умений.

11.3. 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

11.4. Методические указания для обучающихся по выполнению курсового проекта/ 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 по профессиональным учебным дисциплинам и модулям в соответствии с требованиями к уровню подготовки, установленными программой учебной дисциплины, программой подготовки специалиста соответствующего уровня, квалификации;

– применить полученные знания, умения и практический опыт при решении комплексных задач, в соответствии с основными видами профессиональной деятельности по направлению/ специальности/ программе;

– углубить теоретические знания в соответствии с заданной темой;

– сформировать умения применять теоретические знания при решении нестандартных задач;

– сформировать умения работы со специальной литературой, справочной, нормативной и правовой документацией и иными информационными источниками;

– сформировать умения формулировать логически обоснованные выводы, предложения и рекомендации по результатам выполнения работы;

– развить профессиональную письменную и устную речь обучающегося;

– развить системное мышление, творческую инициативу, самостоятельность, организованность и ответственность за принимаемые решения;

– сформировать навыки планомерной регулярной работы над решением поставленных задач.

Структура пояснительной записки курсового проекта/ работы

При выполнении курсовой работы предлагается следующая структура пояснительной записки.

Задание на курсовую работу.

Введение.

1. Постановка задачи, описание предметной области.
2. Анализ и выбор средств реализации проекта.
3. Диаграмма классов для разрабатываемой библиотеки.
4. Программный код классов с комментариями.
5. Интерфейс приложения, разработанный на основе библиотеки классов.

Заключение.

Список использованных источников.

Требования к оформлению пояснительной записки курсового проекта/ работы

Записка к курсовой работе оформляется в соответствии с шаблоном приведенном на сайте ГУАП в разделе Нормативная документация https://guap.ru/regdocs/docs/uch.

11.5. 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.

В процессе выполнения самостоятельной работы у обучающегося формируется целесообразное планирование рабочего времени, которое позволяет ему 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.

11.6. 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, консультаций к курсовой работе, аудиторных занятий для лабораторных и практических работ. Отчеты и практические результаты выполнения всех лабораторных работ должны быть защищены.

11.7. 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».

Итоговая оценка по экзамену формируется как среднеарифметическое оценок, полученных в ходе текущего контроля, и оценки, полученной в результате ответа на экзамене.

Промежуточный контроль успеваемости проводится в соответствии со стандартом организации ГУАП системы менеджмента качества 3.76 «Положение о текущем контроле успеваемости и промежуточной аттестации студентов и аспирантов ГУАП, обучающихся по образовательным программам высшего образования» (https://docs.guap.ru/smk/3.76.pdf).

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