

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В.Г.ШУХОВА»
(БГТУ им. В.Г. Шухова)



РАБОЧАЯ ПРОГРАММА
дисциплины

Объектно-ориентированное программирование

направление подготовки:

09.03.01 «Информатика и вычислительная техника»

Направленность программы (профиль):

Интеллектуальные системы

Квалификация

Бакалавр

Форма обучения

Очная

Институт информационных технологий и управляющих систем

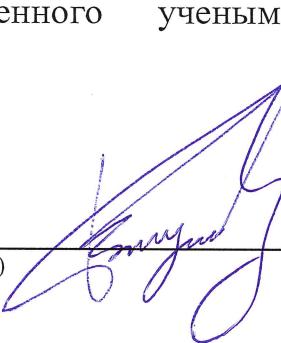
Кафедра программного обеспечения вычислительной техники и
автоматизированных систем

Рабочая программа составлена на основании требований:

- Федерального государственного образовательного стандарта высшего образования – бакалавриат по направлению подготовки 09.03.01 «Информатика и вычислительная техника», утвержденного приказа Минобрнауки России от 19.09.2017 № 929
- учебного плана, утвержденного ученым советом БГТУ им. В.Г. Шухова в 2023 году.

Составитель: ст. преподаватель

(ученая степень и звание, подпись)


(Д.Г. Буханов)

(инициалы, фамилия)

Рабочая программа обсуждена на заседании кафедры

« 21 » 05 2024 г., протокол № 11

Заведующий кафедрой: к.т.н., доцент

(ученая степень и звание, подпись)

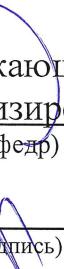

(В.М. Поляков)

(инициалы, фамилия)

Рабочая программа согласована с выпускающей кафедрой программного обеспечения вычислительной техники и автоматизированных систем
(наименование кафедры/кафедр)

Заведующий кафедрой: к.т.н., доцент

(ученая степень и звание, подпись)


(В.М. Поляков)

(инициалы, фамилия)

« 21 » 05 2024 г.

Рабочая программа одобрена методической комиссией института

« 27 » 05 2024 г., протокол № 9

Председатель доцент

(ученая степень и звание, подпись)


(Ю.Д. Рязанов)

(инициалы, фамилия)

1. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Категория (группа) компетенций	Код и наименование компетенции	Код и наименование индикатора достижения компетенции	Наименование показателя оценивания результата обучения при прохождении практики
	ПК-1 Способен разрабатывать требования и проектировать программное обеспечение	ПК-1.1 Анализирует требования к программному обеспечению ПК-1.2 Разрабатывает технические спецификации на программные компоненты и их взаимодействие ПК-1.3 Проектирует программное обеспечение	Знать: подходы к выявлению и анализу требований к разрабатываемому программному обеспечению. Уметь: выполнять декомпозицию предметной области и формировать требования к программному обеспечению. Знать: основные понятия ООП, этапы, методологии разработки объектно-ориентированного программного обеспечения; основы составления технических спецификаций. Навыки: формирования технических спецификаций на программные модули и компоненты. Знать: существующие виды программных архитектур; принципы моделирования объектно-ориентированных программ с использованием языка UML. Уметь: проектировать архитектуру разрабатываемого программного обеспечения. Навыки: использования методологий разработки сложных объектно-ориентированных программ; проектирования и моделирования ООП программ с использованием языка UML.
	ПК-2 Способен разрабатывать программное обеспечение для информационных систем с интеллектуальными компонентами	ПК-2.1 Разрабатывает программное обеспечение обработки информации на основе интеллектуальных технологий	Знать: метрики оценки качества и эффективности программных частей. Уметь: проектировать малосвязные программные части (классы, модули, компоненты, библиотеки); оценивать качество структуры программного обеспечения. Навыки: разработка программных модулей, компонент и библиотек.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

1. Компетенция ПК-1 Способен разрабатывать требования и проектировать программное обеспечение

Данная компетенция формируется следующими дисциплинами.

Стадия	Наименования дисциплины
1.	Алгоритмы и структуры данных
2.	Объектно-ориентированное программирование
3.	Интеллектуальные системы реального времени
4.	Программирование мобильных устройств
5.	Тестирование программных систем
6.	Моделирование систем
7.	Архитектура вычислительных систем
8.	Программирование распределённых систем
9.	Программирование микроконтроллеров
10.	Микропроцессорные системы
11.	Технологии Web-программирования
12.	Программирование мобильной робототехники
13.	Производственная преддипломная практика

2. Компетенция ПК-2 Способен разрабатывать программное обеспечение для информационных систем с интеллектуальными компонентами

Данная компетенция формируется следующими дисциплинами.

Стадия	Наименования дисциплины
1.	Алгоритмы и структуры данных
2.	Объектно-ориентированное программирование
3.	Анализ данных
4.	Методы машинного обучения
5.	Архитектура вычислительных систем
6.	Программирование распределённых систем
7.	Производственная преддипломная практика

3. ОБЪЕМ ДИСЦИПЛИНЫ

Вид учебной работы	Всего часов	Семестр № 3
Общая трудоемкость дисциплины, час	216	216
Контактная работа (аудиторные занятия), в т.ч.:		
лекции	34	34
лабораторные	34	34
практические		
групповые консультации в период теоретического обучения и промежуточной аттестации	5	5
Самостоятельная работа студентов, включая индивидуальные и групповые консультации, в том числе:	143	143
Курсовой проект		
Курсовая работа	36	36
Расчетно-графическое задание		
Индивидуальное домашнее задание		
Самостоятельная работа на подготовку к аудиторным занятиям (лекции, практические занятия, лабораторные занятия)	71	71
Экзамен	36	36

4. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

4.1 Наименование тем, их содержание и объем

Курс 2 Семестр 2

№ п/п	Наименование раздела (краткое содержание)	Объём на тематический раздел по видам учебной нагрузки, час			
		Лекции	Практические занятия	Лабораторные занятия	Самостоятельная работа
1.	Введение в объектно-ориентированное программирование.				
	Программные продукты как сложные системы. Признаки сложности. Назначение объектно-ориентированного программирования.	4	—	4	9
2.	Принципы объектно-ориентированного программирования. Объектно-ориентированное проектирование.				
	Абстрагирование, иерархическая организация, ограничение доступа, модульность. Определение объекта и класса. Объектная декомпозиция, диаграмма классов. Модули. Интерфейсы и реализации. Классы, виды отношений между классами. Наследование, полиморфизм и инкапсуляция. Классы в C++. Синтаксис и особенности C++. Многопоточность. Синхронизация потоков. Мьютексы и их реализация в библиотека STL. Устойчивость, области видимости и типы переменных. Типизация, проблемы приведения типов объектов одной иерархии.	16	—	12	26
3.	Проектирование компонент и модулей. Создание шаблонов классов, стандартные шаблоны STL.				
	Выделение внешних интерфейсов. Динамические ошибки. Исключительные ситуации. Проектирование и разработка структуры исключительных ситуаций. Шаблоны классов, механизм в C++ для его реализации. Изучение библиотек стандартных шаблонов (STL). Контейнеры объектов. Разработка пользовательских контейнеров.	8	—	12	21
4.	Архитектура программного обеспечения. Шаблоны проектирования.				
	Архитектура программного обеспечения. Виды архитектур. Способы формирования архитектур. Предметно-ориентированное проектирование. Рефакторинг. Порождающие шаблоны проектирования, структурные шаблоны проектирования, шаблоны поведения. Принципы SOLID. Тенденции и пути развития ООП.	6	—	6	15
	ВСЕГО:	34	—	34	71

4.2. Содержание практических (семинарских) занятий

Учебным планом не предусмотрено.

4.3. Содержание лабораторных занятий

№ п/п	Наименование раздела дисциплины	Тема лабораторного занятия	Кол-во лекц. часов	Кол- во часов СРС
семестр № 4				
1	Введение в объектно-ориентированное программирование.	Создание приложений в ИСР Microsoft Visual Studio и QT	2	3
2	Принципы объектно-ориентированного программирования. Объектно-ориентированное проектирование.	Модульное программирование. Интерфейсы.	2	3
3	Принципы объектно-ориентированного программирования. Объектно-ориентированное проектирование.	Объектная декомпозиция	4	6
4	Принципы объектно-ориентированного программирования. Объектно-ориентированное проектирование.	Создание классов. Перегрузка операторов	2	3
5	Принципы объектно-ориентированного программирования. Объектно-ориентированное проектирование.	Классы, виды отношений. Наследование.	4	6
6	Принципы объектно-ориентированного программирования. Объектно-ориентированное проектирование.	Потоки в C++	2	3
7	Проектирование компонент и модулей. Создание шаблонов классов, стандартные шаблоны STL.	Обработка исключительных ситуаций в C++	4	6
8	Проектирование компонент и модулей. Создание шаблонов классов, стандартные шаблоны STL.	Создание шаблонов классов в C++.	4	6
9	Проектирование компонент и модулей. Создание шаблонов классов, стандартные шаблоны STL.	Использование стандартной библиотеки шаблонов STL	4	6
10	Архитектура программного обеспечения. Шаблоны проектирования.	Шаблоны проектирования. Разработка программного обеспечения в ИСР QT.	4	6
11	Архитектура программного обеспечения. Шаблоны	Проектирование сложных систем	2	3

проектирования.			
	ИТОГО:	34	51

4.4. Содержание курсового проекта/работы

Целью выполнения курсовой работы является углубление и закрепление студентами знаний основных приемов, методов и принципов работы при решении на ЭВМ задач с использованием языков высокого уровня. Для выполнения работы достаточно знаний основных разделов дисциплины "Объектно-ориентированного программирования". Курсовая работа заключается в разработке модулей подпрограмм и приложений для решения задач заданного класса.

Курсовая работа оформляется в виде пояснительной записи (15-20 стр.), в которой отражаются все этапы создания программного продукта. Обязательными элементами работы являются: объектная модель, диаграмма классов, реализация на языке C++, результаты тестирования.

Примерная тематика курсовых работ

1. Система учета успеваемости студентов в ВУЗе. Программа должна учитывать результаты выполнения студентами различных видов учебной нагрузки (лабораторные работы, РГЗ, курсовые проекты, зачеты, экзамены). Студенты принадлежат различным группам, факультетам, институтам.
2. Редактор UML-диаграмм. Программа должна обеспечить возможность построения UML-диаграмм различных типов. Пользователю должна быть предоставлена возможность сохранения/открытия диаграммы из файла.
3. Программа построения графиков функций. Программа должна строить графики функций различного вида: многочлены, тригонометрические функции и т.п. Графики функций выводятся в специальном поле вывода. Пользователь должен иметь возможность изменять видимый диапазон по осям абсцисс и ординат, изменять коэффициенты отображаемых функций, очищать поле вывода.
4. Программа моделирования движения транспорта. В специальном поле вывода отображается структура улиц населенного пункта. Транспортные средства (автомобили, автобусы и т.п.) представляются условными изображениями. Каждое транспортное средство имеет свой маршрут и скорость. Транспортные средства перемещаются по улицам населенного пункта с учетом своей скорости и модельного времени. Пользователь должен иметь возможность изменять скорость модельного времени, добавлять и удалять транспортные средства, изменять маршрут транспортных средств.
5. Программа для игры в морской бой. Программа должна обеспечивать возможность игры человека с компьютером. На экране отображаются два игровых поля: поле для расстановки кораблей человеком и поле для отметки

наносимых ударов по кораблям противника. Должна быть предусмотрена возможность использования кораблей различного типа: одно-, двух-, трех- и четырехпалубных. Пользователь в режиме расстановки кораблей должен иметь возможность размещения кораблей на игровом поле, перемещения, поворота и удаления кораблей. В режиме игры пользователь в наглядном виде должен информироваться о том, достиг ли цели его выстрел и выстрел противника.

6. Редактор векторных изображений. Программа должна обеспечивать возможность создания векторных изображений с использованием простейших примитивов (линия, прямоугольник, эллипс), заливки произвольных замкнутых областей, масштабирования, изменения цвета, перемещения, сохранения во внешний файл в некотором формате и загрузки изображения из внешнего файла.
7. Программа для построения генеалогического дерева. Программа должна обеспечивать возможность ввода информации о людях, состоящих в родстве друг с другом (пол, дата рождения, место рождения, основной вид деятельности и т.п.). Для каждого человека должна быть возможность указания его родителей, детей. Для регистрации изменений в программе должны быть предусмотрены события различного типа: рождение, смерть, женитьба и т.п. По данным о родственных связях программа должна обеспечить вывода на экран генеалогического дерева.
8. Программа моделирования спортивной игры (футбол). Программа должна обеспечить возможность учета игроков различных команд, моделирования матчей команд, учета результатов матчей в различных турнирах. Каждая команда имеет отличительные свойства (например, настрой, умение нападать, умение защищаться), которые влияют на результаты матчей с участием этой команды. Каждый игрок характеризуются некоторыми атрибутами, которые влияют на его поведение во время матча (например, скорость, точность удара, выносливость и т.п.).
9. Программа моделирования спортивной игры (волейбол). Программа должна обеспечить возможность учета игроков различных команд, моделирования матчей команд, учета результатов матчей в различных турнирах. Каждая команда имеет отличительные свойства (например, настрой, умение нападать, умение защищаться), которые влияют на результаты матчей с участием этой команды. Каждый игрок характеризуются некоторыми атрибутами, которые влияют на его поведение во время матча (например, скорость, точность удара, выносливость и т.п.).
10. Программа моделирования спортивной игры (баскетбол). Программа должна обеспечить возможность учета игроков различных команд, моделирования

матчей команд, учета результатов матчей в различных турнирах. Каждая команда имеет отличительные свойства (например, настрой, умение нападать, умение защищаться), которые влияют на результаты матчей с участием этой команды. Каждый игрок характеризуются некоторыми атрибутами, которые влияют на его поведение во время матча (например, скорость, точность удара, выносливость и т.п.).

4.5. Содержание расчетно-графического задания, индивидуальных домашних заданий

Учебным планом не предусмотрено

5. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ПРОВЕДЕНИЯ ТЕКУЩЕГО КОНТРОЛЯ, ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

5.1. Реализация компетенций

- 1. Компетенция ПК-1.** Способен разрабатывать требования и проектировать программное обеспечение

Наименование индикатора достижения компетенции	Используемые средства оценивания
ПК-1.1 Анализирует требования к программному обеспечению	защита лабораторных работ, экзамен
ПК-1.2 Разрабатывает технические спецификации на программные компоненты и их взаимодействие	защита лабораторных работ
ПК-1.3 Проектирует программное обеспечение, в том числе для беспилотных авиационных систем	защита лабораторных работ, экзамен

- 2. Компетенция ПК-2.** Способен разрабатывать программное обеспечение для информационных систем с интеллектуальными компонентами

Наименование индикатора достижения компетенции	Используемые средства оценивания
ПК-3.1 Разрабатывает программное обеспечение обработки информации на основе интеллектуальных технологий	защита лабораторных работ

5.2. Типовые контрольные задания для промежуточной аттестации

5.2.1. Перечень контрольных вопросов (типовых заданий) для экзамена

№ п/п	Наименование раздела дисциплины	Содержание вопросов (типовых заданий)
1.	Введение в объектно-ориентированное программирование (ПК-1).	Программные продукты как сложные системы. Признаки сложных систем. Декомпозиция. Причины сложности программных систем. Объект. Что не является объектом, подходы к выделению объектов. Понятие модуля. Интерфейс. Интерфейс, класс, тип.

2.	<p>Принципы объектно-ориентированного программирования Объектно-ориентированное проектирование (ПК-1).</p>	<p>Классы. Абстракции. Этапы разработки программных средств с использованием объектно-ориентированного подхода. Объектно-ориентированные языки программирования. Принципы объектно-ориентированного представления программных систем (модульность, иерархическая организация). Полиморфизм. Инкапсуляция. Общая характеристика объектов. Объектная декомпозиция. Объектная модель. Диаграмма классов. Виды отношений между классами. Общая характеристика классов. Обработка исключительных ситуаций в C++. Обработка общих исключительных ситуаций в C++. Виды отношений между объектами. Дополнительные принципы ООП. Многопоточность. Синхронизация потоков. Мьютексы и их реализация в библиотека STL. Устойчивость, области видимости и типы переменных. Типизация, проблемы приведения типов объектов одной иерархии. Многопоточное программирование в C++. Класс Thread</p>
3.	<p>Проектирование компонент и модулей. Создание шаблонов классов, стандартные шаблоны STL (ПК-1).</p>	<p>Виды методов в C++. Раннее и позднее связывание. Виртуальные методы. Таблица виртуальных методов. Виртуальные методы. Практическое применение. Конструкторы и инициализация объектов. Конструкторы и методы создания экземпляра класса. Библиотека стандартных классов (STL). Преобразование типов в C++. dynamic_cast, static_cast, reinterpret_cast, const_cast. Шаблоны классов, template. Контейнеры STL. Основные методы. Стандартная библиотека STL (map, list). Стандартная библиотека STL (vector, stack). Стандартная библиотека STL (set, queue).</p>

4.	Архитектура программного обеспечения. Шаблоны проектирования (ПК-2).	Принципы SOLID (SO). Принципы SOLID (LI). Принципы SOLID (ID). Шаблоны проектирования. Порождающие шаблоны. Шаблоны проектирования. Структурные шаблоны. Шаблоны проектирования. Шаблоны поведения. Шаблон проектирования. Итератор. Шаблон проектирования. Фабричный метод. Шаблон проектирования. Одиночка. Шаблон проектирования. Абстрактная фабрика. Шаблон проектирования. Прототипирование. Архитектура программного обеспечения и ее виды. Тестирование. TDD. Тесты, как способ формирования архитектуры. Рефакторинг. Виды рефакторинга. Составление методов. Рефакторинг. Виды рефакторинга. Перенос функциональности между объектами. Объектная структура предметной области. DDD. Объект-значение и агрегат. DDD. Инструменты проектирования. Диаграмма состояний. Инструменты проектирования. Диаграмма взаимодействия (диаграмма последовательностей). Метрики оценки программного кода.
----	--	---

5.3. Типовые контрольные задания (материалы) для текущего контроля в семестре

Текущий контроль осуществляется в течение семестра в форме защиты лабораторных работ.

В методических указаниях к выполнению лабораторных работ по дисциплине представлен перечень лабораторных работ, обозначены цель и задачи, представлены индивидуальные варианты заданий и перечень контрольных вопросов.

Захиста лабораторной работы проводиться в форме устного опроса студента и направлена на проверку степени усвоения материала и понимания теоретических сведений, используемых в процессе выполнения работы; для защиты необходимо представить в печатной (рукописной) форме отчет по лабораторной работе, выполненный самостоятельно и в соответствии со всеми требованиями, приведёнными в методических указаниях к выполнению лабораторных работ. Примерные перечень контрольных вопросов для защиты лабораторных работ приведен в таблице:

Тематика лабораторной работы	Контрольные вопросы
Лабораторная работа №1. Создание приложений в ИСР Microsoft Visual Studio и QT (ПК-1)	1. Структура приложения в среде разработки MS Visual Studio или QT. 2. Типы приложений, создаваемые в MS Visual Studio или QT. 3. Назначение и основные свойства стандартных компонентов Windows Forms

	<p>4. Что такое проект в MS Visual Studio или QT? Какие файлы входят в проект?</p> <p>5. Отладка приложений в MS Visual Studio.</p>
Лабораторная работа №2. Модульное программирование. Интерфейсы (ПК-1)	<p>1.Что такое модуль?</p> <p>2.Какие бывают типы модулей в C++?</p> <p>3.Структура данных типа «запись» (Struct).</p> <p>4.Интерфейс.</p> <p>5. Причины сложности при разработки ПО.</p>
Лабораторная работа №3. Объектная декомпозиция (ПК-1)	<p>1.Понятие объекта.</p> <p>2.Объектная декомпозиция.</p> <p>3. Отличия объектной декомпозиции от алгоритмической.</p> <p>4. Как объекты взаимодействуют друг с другом?</p> <p>5. Виды отношений между объектами.</p>
Лабораторная работа №4. Создание классов. Перегрузка операторов (ПК-1)	<p>1. Понятие класса.</p> <p>2. Абстрагирование.</p> <p>3. Принципы ООП.</p> <p>4. Способы перегрузки операторов.</p> <p>5. Простой полиморфизм.</p>
Лабораторная работа №5. Классы, виды отношений. Наследование. (ПК-1)	<p>1.Полиморфизм.</p> <p>2. Виды полиморфизма.</p> <p>3. Виды отношения между классами</p> <p>4. Виды методов.</p> <p>5. Концептуальное отличие методы от функции.</p>
Лабораторная работа №6. Потоки в C++ (ПК-1)	<p>1.Дополнительные принципы ООП.</p> <p>2. Виды параллельных вычислений.</p> <p>3. Вычислительный поток.</p> <p>4. Синхронизация потоков.</p> <p>5. Особенности запуска потоков использую модуль <code>std::thread</code></p>
Лабораторная работа №7. Обработка исключительных ситуаций в C++ (ПК-1)	<p>1.Работа с ошибками.</p> <p>2. Проблемы при отладке ПО.</p> <p>3. Исключительные ситуации.</p> <p>4. Типы ошибок.</p> <p>5. Иерархия исключительных ситуаций.</p> <p>6. Принципы работы с исключениями в C++.</p>
Лабораторная работа №8. Создание шаблонов классов в C++ (ПК-2)	<p>1.Шаблоны классов.</p> <p>2. Отношение инстанцирования.</p> <p>3. Виды аргументов при реализации шаблона в C++</p> <p>4. Шаблоны функций.</p>
Лабораторная работа №9. Использование стандартной библиотеки шаблонов STL (ПК-2)	<p>1.Библиотека стандартных классов (STL).</p> <p>2.Преобразование типов в C++. <code>dynamic_cast</code>, <code>static_cast</code>, <code>reinterpret_cast</code>, <code>const_cast</code>.</p> <p>3. Шаблоны классов, <code>template</code>.</p> <p>4. Контейнеры STL. Основные методы.</p> <p>5. Стандартная библиотека STL (<code>map</code>, <code>list</code>).</p> <p>6. Стандартная библиотека STL (<code>vector</code>, <code>stack</code>).</p> <p>7. Стандартная библиотека STL (<code>set</code>, <code>queue</code>).</p>
Лабораторная работа №10. Шаблоны проектирования. Разработка программного обеспечения в ИСР QT (ПК-2).	<p>1.Умные указатели.</p> <p>2. Использование деструкторов.</p> <p>3. Принципы работы с QTForm</p> <p>4. Структура компонент QT.</p>

	5. Слоты и сигналы QT.
Лабораторная работа №11. Проектирование сложных систем (ПК-2)	<ol style="list-style-type: none"> 1. Архитектура программного обеспечения и ее виды. 2. Тесты, как способ формирование архитектуры. 3. Рефакторинг. Виды рефакторинга. 4. Предметно-ориентированное проектирование. 5. Инструменты проектирования. Виды диаграмм UML. 6. Метрики оценки программного кода.

Примерный перечень тестовых заданий для защиты лабораторных работ приведен в таблице:

№ п/п	Наименование раздела дисциплины	Содержание вопросов (типовых заданий)
1	Введение в объектно- ориентированное программирование (ПК-1).	<ol style="list-style-type: none"> 1. Представление предметной области в виде объектов взаимодействующих между собой посредством передачи сообщений называется: <ol style="list-style-type: none"> а) алгоритмическая декомпозиция; б) объектная декомпозиция; в) функциональная декомпозиция. 2. При каскадной модели разработки программного обеспечения выделяют следующие этапы: <ol style="list-style-type: none"> а) анализ, проектирование, реализация; б) анализ, реализация; в) вертикальные срезы функций системы. 3. Какими по умолчанию объявляются методы класса? <ol style="list-style-type: none"> а) private; б) public; в) protected. 4. Укажите не правильный идентификатор переменной: <ol style="list-style-type: none"> а) 10xd; б) xd_10; в) xd10. 5. Какой из перечисленных типов является встроенными типами языка C++? <ol style="list-style-type: none"> а) real; б) double; в) integer. 6. Точкой входа в программу на языке C++ является: <ol style="list-style-type: none"> а) первая описанная функция в главном модуле; б) функция main; в) любая функция. 7. Поля и методы доступные экземплярам текущего класса и его потомкам? <ol style="list-style-type: none"> а) private, protected; б) public, protected; в) private, public.

8. После выполнения программы, чему равняется переменная a?

```
int a;
for (int i=1; i<10; i++){
    static int j = 0;
    j++;
    a = j;
}
```

a) 9;
б) 10;
в) 0.

9. Что из перечисленного для языка C++ не верно?

- а) struct является полноценным типом;
- б) все поля и методы по умолчанию в struct являются публичными;
- в) нельзя определить private методы в struct.

10. После выполнения кода чему равняется значение указателя a?

```
void test(int *a){
    *a = 3;
};

int main() {
    int *a = new int(1);
    test(a);
    std::cout<<*a;

    return 0;
}
```

a) 3;
б) 1;
в) ошибка при компиляции.

11. После выполнения кода чему равняется значение указателя a?

```
void test(const int *a){
    *a = 3;
};

int main() {
    int *a = new int(1);
    test(a);
    std::cout<<*a;

    return 0;
}
```

a) 3;
б) 1;
в) ошибка при компиляции.

12. После выполнения кода чему равняется a?

	<pre> void test(int &a) { a = 3; }; int main() { int a = 1; test(a); std::cout << a; return 0; } a) 3; б) 1; в) ошибка при компиляции. </pre> <p>13. После выполнения кода чему равняется значение указателя a?</p> <pre> void test(int *a) { int *b = new int(2); a = b; }; int main() { int *a = new int(1); test(a); std::cout << *a; return 0; } a) 2; б) 1; в) ошибка при компиляции. </pre>
2	<p>Принципы объектно-ориентированного программирования Объектно-ориентированное проектирование (ПК-1).</p> <ol style="list-style-type: none"> Основным принципом ООП не является: <ul style="list-style-type: none"> а) абстрагирование; б) иерархическая организация; в) адаптивность. Какой вид отношения между двумя классами предполагает физическое включения одного объекта в другой? <ul style="list-style-type: none"> а) наследование; б) композиция; в) зависимость. Интерфейс – это? <ul style="list-style-type: none"> а) доступные элементы из вне абстракции; б) недоступные элементы из вне абстракции; в) класс. Иерархия общее – частное (is-a) используется при: <ul style="list-style-type: none"> а) наследовании; б) зависимостии; в) агрегации.

		<p>5. Иерархия часть – целое (part of) используется при:</p> <ol style="list-style-type: none"> наследовании; зависимости; агрегации. <p>6. К дополнительным принципам не относится:</p> <ol style="list-style-type: none"> абстрагирование; типизация; параллелизм; устойчивость. <p>7. Как неправильно проинициализировать константное поле в объекте?</p> <ol style="list-style-type: none"> <pre>class Test { const int a; public: Test() { a = 2; } };</pre> <pre>class Test { const int a; public: Test():a(2){} };</pre> <pre>class Test { const int a; public: Test():a{2}{} };</pre> <p>8. Чем отличается сигнатура метод от функции?</p> <ol style="list-style-type: none"> ничем; у методы класса, есть явный или не явный аргумент, который является «объектом» над которым он вызван; функцию нельзя вызвать без параметров. <p>9. UML диаграмма классов содержит?</p> <ol style="list-style-type: none"> объекты и их классы; классы и их отношения; модули и их взаимодействия. <p>10. UML описания класса содержит</p> <ol style="list-style-type: none"> название класса и его функции; название класса, поля и его методы; перечень создаваемых объектов.
3	Проектирование компонент и модулей.	<p>1. Какой тип имеет переменная <code>auto i = 1;</code> в C++?</p> <ol style="list-style-type: none"> double; int;

<p>Создание шаблонов классов, стандартные шаблоны STL (ПК-1, ПК-2).</p>	<p>в) char.</p> <p>2. Каким набором методов должен обладать «Итератор»?</p> <p>а) конструктор, деструктор, переход на следующий элемент, получение значения текущего элемента, проверка на конец.</p> <p>б) конструктор, деструктор, переход на следующий элемент;</p> <p>в) не требует специальных методов.</p> <p>3. Чем является std::vector?</p> <p>а) шаблоном;</p> <p>б) классом;</p> <p>в) объектом.</p> <p>4. Какие типы компонентов не являются частью STL библиотеки?</p> <p>а) контейнер;</p> <p>б) алгоритм;</p> <p>в) модульные тесты.</p> <p>5. Какое утверждение не верно?</p> <p>а) модуль является единицей компиляции;</p> <p>б) модуль является физическим контейнером элементов программы;</p> <p>в) модуль может содержать только одну функцию.</p> <p>6. Что выведет следующая программа ?</p> <pre>char *s = "C++"; s++; std::cout << s;</pre> <p>а) C++;</p> <p>б) ++;</p> <p>в) ошибку.</p> <p>7. Сколько стандартных потоков определяет Библиотека iostream?</p> <p>а) 0;</p> <p>б) 1;</p> <p>в) 3.</p> <p>8. Что будет выведено на экран?</p> <pre>std::vector<int> a = { 1, 2, 3, 4 }; for (auto el: a) { std::cout << el; }</pre> <p>а) ошибка;</p> <p>б) 1234;</p> <p>в) 4.</p> <p>9. Что будет выведено на экран?</p> <pre>std::vector<int> a = { 1, 2, 3, 4 }; std::vector<int> b = { 4, 3, 2, 1 }; std::swap(a,b);</pre>
---	---

		<p>for (auto el: a) { std::cout << el; } a) ошибка; б) 1234; в) 4321.</p> <p>10. Какую структуру данных описывает шаблон <code>map<>?</code> а) вектор; б) карту; в) ассоциативный массив.</p> <p>11. Идеями гибких методологий управления процессом разработки не являются? а) готовность к изменениям важнее следования первоначальному плану; б) люди и взаимодействие важнее процессов и инструментов; в) документация важнее работающего процесса.</p>
4	<p>Архитектура программного обеспечения. Шаблоны проектирования (ПК-2).</p>	<p>1. Какой принцип SOLID в следующем коде нарушен?</p> <pre>class A{ public: virtual void action(); }; class B: public A{ public: void action(); }; class C: public A{ public: void action(int a); };</pre> <p>а) единственной ответственности; б) Барбары Лисков; в) инверсии зависимостей.</p> <p>2. Каким свойством не должен обладать автоматический тест? а) повторяемостью; б) простотой; в) гибкостью.</p> <p>3. Разделяет создание сложного объекта и инициализацию его состояния так, что одинаковый процесс построения может создать объекты с разным состоянием. О каком паттерне идёт речь? а) строитель; б) наблюдатель; в) хранитель.</p>

4. Определяет несколько видов объектов, чтобы при создании использовать объект-макет и создаёт новые объекты, копируя макет. О каком паттерне идёт речь?
- а) итератор;
 - б) прототип;
 - в) интерпретатор;
5. Что из этого описывает порождающие шаблоны?
- а) определяют взаимодействие между объектами, увеличивая таким образом его гибкость;
 - б) изменяют интерфейс уже существующих объектов или его реализацию;
 - в) шаблоны проектирования, которые абстрагируют процесс создания.
6. Что из этого не относится к видам паттернов проектирования?
- а) примитивные;
 - б) порождающие;
 - в) структурные.
7. Гарантирует, что класс имеет только один экземпляр и представляет глобальную точку доступа к нему.
- а) фасад;
 - б) строитель;
 - в) одиночка.
8. Позволяет осуществлять последовательную итерацию, не вдаваясь во внутреннее устройство коллекции?
- а) итератор;
 - б) строитель;
 - в) наблюдатель.
9. Что не является плохой практикой при разработке ПО?
- а) излишняя сложность;
 - б) отсутствие структуры;
 - в) рефакторинг.
10. Шаблон проектирования – это ...
- а) общее решение некоторой повторяющейся проблемы в определенном контексте;
 - б) экземпляр объекта, который выдает себя за группу самостоятельных экземпляров;
 - в) алгоритмы, основанные на возможности повторного использования преобразований и возможности модификации существующей системы посредством добавления новых преобразований.
11. Принцип открытости закрытости предполагает
- а) много интерфейсов, специально предназначенных для клиентов, лучше, чем один интерфейс общего назначения;
 - б) что программные сущности должны быть открыты для расширения, но закрыты для модификации

	<p>в) необходимо инкапсулировать состояние объектов.</p> <p>12. Рефакторинг – это?</p> <p>а) оптимизация выполнения кода;</p> <p>б) добавление новых функций к системе;</p> <p>в) модификация кода с целью повышения его понимания.</p>
--	---

Критерии оценки лабораторной работы: лабораторная работа считается защищенной, если студент выполнил задание к работе полностью, правильно ответил на тестовые вопросы или во время собеседования правильно ответил на вопросы, заданные преподавателем, выполнил дополнительные задания.

5.4. Описание критериев оценивания компетенций и шкалы оценивания

При промежуточной аттестации в форме экзамена используется следующая шкала оценивания: 2 – неудовлетворительно, 3 – удовлетворительно, 4 – хорошо, 5 – отлично.

При промежуточной аттестации в форме зачета используется следующая шкала оценивания: зачтено, не зачтено.

Критериями оценивания достижений показателей являются:

Наименование показателя оценивания результата обучения по дисциплине	Критерий оценивания
Знания	Знание терминов, определений, понятий
	Знание основных принципов и подходов, как при ОО проектировании, так и при ОО программировании
	Объем освоенного материала
	Полнота ответов на вопросы
	Четкость изложения и интерпретации знаний
Умения	Умение решать стандартные профессиональные задачи с применением принципов объектно-ориентированного программирования
	Умение использовать теоретические знания при декомпозиции задач предметной области
	Умение проверять решение и анализировать результаты
Навыки	Владение навыками проектирования объектно-ориентированных систем
	Владение навыками оценки и выявления требований к разработке программного продукта.
	Самостоятельность выполнения исследований объектов профессиональной деятельности.

Оценка преподавателем выставляется интегрально с учётом всех показателей и критериев оценивания.

Оценка сформированности компетенций по показателю Знания.

Критерий	Уровень освоения и оценка			
	2	3	4	5

Знание терминов, определений, понятий	Не знает терминов и определений	Знает термины и определения, но допускает неточности формулировок	Знает термины и определения	Знает термины и определения, может корректно сформулировать их самостоятельно
Знание основных принципов и подходов, как при ОО проектировании, так и при ОО программировании и	Не знает основные принципы и подходы, как при ОО проектировании, так и при ОО программировании	Знает основные принципы ООП.	Знает основные принципы ООП, их интерпретирует и использует	Знает основные принципы ООП, может самостоятельно их сформулировать и использовать
Объем освоенного материала	Не знает значительной части материала дисциплины	Знает только основной материал дисциплины, не усвоил его деталей	Знает материал дисциплины в достаточном объеме	Обладает твердым и полным знанием материала дисциплины, владеет дополнительными знаниями
Полнота ответов на вопросы	Не дает ответы на большинство вопросов	Дает неполные ответы на все вопросы	Дает ответы на вопросы, но не все - полные	Дает полные, развернутые ответы на поставленные вопросы
Четкость изложения и интерпретации знаний	Излагает знания без логической последовательности	Излагает знания с нарушениями в логической последовательности	Излагает знания без нарушений в логической последовательности	Излагает знания в логической последовательности, самостоятельно их интерпретируя и анализируя
	Не иллюстрирует изложение поясняющими схемами, рисунками и примерами	Выполняет поясняющие схемы и рисунки небрежно и с ошибками	Выполняет поясняющие схемы и рисунки корректно и понятно	Выполняет поясняющие рисунки и схемы точно и аккуратно, раскрывая полноту усвоенных знаний
	Неверно излагает и интерпретирует знания	Допускает неточности в изложении и интерпретации знаний	Грамотно и по существу излагает знания	Грамотно и точно излагает знания, делает самостоятельные выводы

Оценка сформированности компетенций по показателю Умения.

Критерий	Уровень освоения и оценка			
	2	3	4	5
Умение решать стандартные профессиональные задачи с применением принципов объектно-ориентированного программирования	Не умеет решать стандартные профессиональные задачи с применением принципов ООП	Допускает неточности в решении стандартных профессиональных задач с применением принципов ООП	Умеет решать стандартные профессиональные задачи с применением принципов ООП	Безошибочно решает стандартные профессиональные задачи с применением принципов ООП
Умение использовать теоретические	Не умеет использовать теоретические	Умеет выполнять ОО декомпозицию предметной	Умеет выполнять ОО декомпозицию	Умеет применять теоретические знания в полной

знания при декомпозиции задач предметной области	знания для выбора методики решения профессиональных задач	области, но затрудняется проектировать последующие решения.	предметной области, может выполнять проектирование последующего решения задачи, но затрудняется применять общепринятые практики при разработке систем, такие как SOLID, DRY, Паттерны проектирования и др.	мере при решении задач проектировании и реализации ОО систем
--	---	---	--	--

Оценка сформированности компетенций по показателю Навыки.

Критерий	Уровень освоения и оценка			
	2	3	4	5
Владение навыками проектирования объектно-ориентированных систем	Не владеет навыками проектирования программных систем	Не достаточно хорошо владеет навыками проектирования при разработке программного обеспечения	Владеет навыками проектирования программных систем	Профессионально владеет навыками проектирования программных систем
Владения навыками оценки и выявления требований к разработке программного продукта.	Не может выполнить декомпозицию предметной области ни в каком виде.	Не достаточно качественно выполняет исследования объектов предметной области, а также не качественно производит выявление требований к разрабатываемой системе	Не достаточно качественно выполняет исследования объектов профессиональной деятельности, допускает и исправляет ошибки самостоятельно	Качественно выполняет оценку и выявления требований к разработке программного продукта
Самостоятельность выполнения исследований объектов профессиональной деятельности	Не может самостоятельно выполнять исследования объектов профессиональной деятельности	Выполняет исследования объектов профессиональной деятельности с посторонней помощью	При выполнении исследования объектов профессиональной деятельности иногда требуется посторонняя помощь	Самостоятельно выполняет исследования объектов профессиональной деятельности

6. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ И УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ

6.1. Материально-техническое обеспечение

№	Наименование специальных помещений и помещений для самостоятельной работы	Оснащенность специальных помещений и помещений для самостоятельной работы
1	Учебная аудитория для проведения лекционных занятий	Специализированная мебель. Мультимедийная установка, экран, доски
2	Учебная аудитория для проведения лабораторных занятий	Специализированная мебель. Компьютеры на базе процессоров Intel или AMD.
3	Читальный зал библиотеки для самостоятельной работы	Специализированная мебель. Компьютерная техника, подключенная к сети интернет и имеющая доступ в электронно-образовательную среду

6.2. Лицензионное и свободно распространяемое программное обеспечение

№	Перечень лицензионного программного обеспечения.	Реквизиты подтверждающего документа
1.	Microsoft Windows 10 Корпоративная	(Соглашение Microsoft Open Value Subscription V9221014 Соглашение действительно с 01.11.2020 по 31.10.2023). Договор поставки ПО № 128-21 от 30.10.2021.
2.	Microsoft Office Professional Plus 2016	(Соглашение Microsoft Open Value Subscription V9221014 Соглашение действительно с 01.11.2020 по 31.10.2023). Договор поставки ПО № 128-21 от 30.10.2021.
3.	Kaspersky Endpoint Security «Стандартный Russian Edition»	Сублицензионный договор № 102 от 24.05.2018. Срок действия лицензии до 19.08.2020 Гражданко-правовой Договор (Контракт) № 27782 «Поставка продления права пользования (лицензии) Kaspersky Endpoint Security от 03.06.2020. Срок действия лицензии 19.08.2022г.
4.	Google Chrome	Свободно распространяемое ПО согласно условиям лицензионного соглашения
5.	Среды программирования Dev C++ , CodeBlocks, Visual Studio Community Edition	Свободно распространяемое ПО согласно условиям лицензионного соглашения

6.3. Перечень учебных изданий и учебно-методических материалов

1. Баженова, И. Ю. Введение в программирование : учебное пособие / И. Ю. Баженова, В. А. Сухомлин. — 3-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. — 326 с. [Электронный ресурс].
2. Леоненков, А. В. Объектно-ориентированный анализ и проектирование с использованием UML и IBM Rational Rose : учебное пособие / А. В. Леоненков. — 3-е изд. — Москва : Интернет-Университет Информационных

- Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. — 317 с. [Электронный ресурс].
3. Мейер, Б. Почувствуй класс. Учимся программировать хорошо с объектами и контрактами : учебник / Б. Мейер ; под редакцией В. А. Биллига. — 3-е изд. —Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. — 775 с. [Электронный ресурс].
 4. Кауфман, В. Ш. Языки программирования. Концепции и принципы / В. Ш. Кауфман. — 2-е изд. — Саратов : Профобразование, 2019. — 464 с.
 5. Мейер, Б. Основы объектно-ориентированного проектирования : учебник / Б. Мейер. — 3-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. — 751 с. [Электронный ресурс].
 6. Фридман, А. Л. Язык программирования С++ : учебное пособие / А. Л. Фридман. — 3-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. — 217 с. [Электронный ресурс].
 7. Фаулер, М. Шаблоны корпоративных приложений : пер. с англ. / М. Фаулер; при участии Д. Райс [и др.]. - Москва : Вильямс, 2010. - 543 с.
 8. Хорев П. Б. Технологии объектно-ориентированного программирования : учеб. пособие / П. Б. Хорев. – 2-е изд., стер. – М. : Академия, 2008. – 448 с. – (Высшее профессиональное образование). – ISBN 978-5-7659-5262-5
 9. Лаптев В. В. С++. Объектно-ориентированное программирование : учеб. пособие / В. В. Лаптев. - Санкт-Петербург : ПИТЕР, 2008. - 457 с
 10. Страуструп, Б. Язык программирования С++ / Б. Страуструп ; пер. с англ.: С. Анисимов, М. Кононов ; ред.: Ф. Андреев, А. Ушаков. - Спец. изд. - Москва : БИНОМ, 2005. - 1098 с.
 11. Буч, Г. Язык UML. Руководство пользователя / Г. Буч, Д. Рамбо, А. Джекобсон. - 2-е изд., стер. - Москва : ДМК пресс, 2004. - 428 с. - (Объектно-ориентированные технологии в программировании).

6.4 Перечень интернет ресурсов, профессиональных баз данных, информационно-справочных систем

1. Электронная библиотека (на базе ЭБС «БиблиоТех») — Режим доступа: <http://ntb.bstu.ru>
2. Электронно-библиотечная система IPRbooks — Режим доступа: <http://www.iprbookshop.ru>
3. Электронно-библиотечная система «Университетская библиотека ONLINE» — Режим доступа: <http://www.biblioclub.ru/>
4. Документация по Visual C++. — Режим доступа: <https://docs.microsoft.com/ru-ru/cpp/>
5. QT Creator. — Режим доступа: <https://www.qt.io/>