

**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):

Вычислительные машины, комплексы, системы и сети

Квалификация

Бакалавр

Форма обучения

Очная

Институт энергетики, информационных технологий и управляющих систем

Кафедра программного обеспечения вычислительной техники и  
автоматизированных систем

Белгород 2019

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 2019 году.

Составитель : \_\_\_\_\_ (Буханов Д.Г.)  
(ученая степень и звание, подпись) (инициалы, фамилия)

Рабочая программа обсуждена на заседании кафедры

« 18 » \_\_\_\_\_ 05 \_\_\_\_\_ 2019 г., протокол № 10

Заведующий кафедрой: к.т.н., доцент \_\_\_\_\_ (Поляков В.М.)  
(ученая степень и звание, подпись) (инициалы, фамилия)

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

Заведующий кафедрой: к.т.н., доцент \_\_\_\_\_ (Поляков В.М.)  
(ученая степень и звание, подпись) (инициалы, фамилия)

« 18 » \_\_\_\_\_ 05 \_\_\_\_\_ 2019 г.

Рабочая программа одобрена методической комиссией института

« 18 » \_\_\_\_\_ 05 \_\_\_\_\_ 2019 г., протокол № 9

Председатель к.т.н., доцент \_\_\_\_\_ (Семернин А.Н.)  
(ученая степень и звание, подпись) (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 по дисциплине
Профессиональные компетенции	ПКВ-1. Способен осуществлять концептуальное, функциональное и логическое проектирование систем среднего и крупного масштаба и сложности.	ПКВ-1.1. Знать: методы концептуального, функционального и логического проектирования систем среднего и крупного масштаба и сложности	<ul style="list-style-type: none"> <li>– особенности объектно-ориентированного проектирования систем;</li> <li>– основы объектно-ориентированного программирования;</li> <li>– основы технологии объектно-ориентированной декомпозиции программных систем;</li> <li>– основные шаблоны проектирования.</li> </ul>
		ПКВ-1.2. Уметь: разрабатывать бизнес-требования к системе, определять цели создания системы, разрабатывать концепции системы	<ul style="list-style-type: none"> <li>– разрабатывать объектную модель и диаграмму классов предметной области;</li> <li>– проектировать и выделять требования к разрабатываемой системе;</li> </ul>
		ПКВ-1.3. Владеть: навыками разработки технического задания на систему	<ul style="list-style-type: none"> <li>– навыками проектного документирования систем среднего и крупного масштаба сложности.</li> </ul>
	ПКВ-3. Способен разрабатывать требования и проектировать программное обеспечение	ПКВ-3.1. Знать: виды требований к программному обеспечению и способы их документирования, методы и технологии проектирования программного обеспечения	<ul style="list-style-type: none"> <li>– виды требований к программному обеспечению;</li> <li>– методы определения требований к программному продукту.</li> </ul>
		ПКВ-3.2. Уметь: разрабатывать технические спецификации на программные компоненты и их взаимодействие	<ul style="list-style-type: none"> <li>– использовать современные методы объектно-ориентированного программирования при проектировании программных систем разного уровня сложности;</li> <li>– составлять спецификации к</li> </ul>

			программным частям (функция, класс, модуль, компонент)
		ПКВ-3.3. Владеть: навыками проектирования программного обеспечения	– методами декомпозиции сложных систем; – методами проектирования систем с использованием классических шаблонов проектирования; – языком объектно-ориентированного программирования C++.

## 2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

**1. Компетенция ПКВ-1.** Способен осуществлять концептуальное, функциональное и логическое проектирование систем среднего и крупного масштаба и сложности.

Данная компетенция формируется следующими дисциплинами.

Стадия	Наименования дисциплины <sup>1</sup>
1.	Системное моделирование
2.	Объектно-ориентированное программирование
3.	Проектирование и управление вычислительными сетями
4.	Проектирование программно-аппаратных комплексов
5.	Спецификация, архитектура и проектирование программных систем
6.	Разработка приложений для OS iOS
7.	Технологии разработки программных систем
8.	Основы искусственного интеллекта
9.	Мультиагентные системы
10.	Технологии Web-программирования
11.	Проектирование клиент-серверных приложений
12.	Архитектура и программирование распределённых вычислительных систем
13.	Программирование высокопроизводительных систем
14.	Разработка приложений для OS Android
15.	Разработка приложений для OS iOS
16.	Производственная преддипломная практика

**2. Компетенция ПКВ-3.** Способен разрабатывать требования и проектировать программное обеспечение

<sup>1</sup> В таблице должны быть представлены все дисциплины и(или) практики, которые формируют компетенцию в соответствии с компетентностным планом. Дисциплины и(или) практики указывать в порядке их изучения по учебному плану.

Данная компетенция формируется следующими дисциплинами.

Стадия	Наименования дисциплины <sup>2</sup>
1.	Алгоритмы и структуры данных
2.	Проектирование программно-аппаратных комплексов
3.	Объектно-ориентированное программирование
4.	Надёжность вычислительных систем
5.	Теория надёжности
6.	Спецификация, архитектура и проектирование программных систем
7.	Разработка приложений для OS iOS
8.	Технологии разработки программных систем
9.	Основы искусственного интеллекта
10.	Мультиагентные системы
11.	Технологии Web-программирования
12.	Проектирование клиент-серверных приложений
13.	Архитектура и программирование распределённых вычислительных систем
14.	Программирование высокопроизводительных систем
15.	Разработка приложений для OS Android
16.	Разработка приложений для OS iOS
17.	Производственная преддипломная практика

---

<sup>2</sup> В таблице должны быть представлены все дисциплины и(или) практики, которые формируют компетенцию в соответствии с компетентностным планом. Дисциплины и(или) практики указывать в порядке их изучения по учебному плану.

### 3. ОБЪЕМ ДИСЦИПЛИНЫ

Общая трудоемкость дисциплины составляет 6 зач. единиц, 216 часов.

Форма промежуточной аттестации

экзамен

(экзамен, дифференцированный зачет, зачет)

Вид учебной работы <sup>3</sup>	Всего часов	Семестр № 3
Общая трудоемкость дисциплины, час	216	216
<b>Контактная работа (аудиторные занятия), в т.ч.:</b>	73	73
лекции	34	34
лабораторные	34	34
практические		
групповые консультации в период теоретического обучения и промежуточной аттестации <sup>4</sup>	5	5
<b>Самостоятельная работа студентов, включая индивидуальные и групповые консультации, в том числе:</b>	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)	107	107
Форма промежуточная аттестация		Экзамен

<sup>3</sup> в соответствии с ЛНА предусматривать

- не менее 0,5 академического часа самостоятельной работы на 1 час лекций,
- не менее 1 академического часа самостоятельной работы на 1 час лабораторных и практических занятий,
- 36 академических часов самостоятельной работы на 1 экзамен
- 54 академических часов самостоятельной работы на 1 курсовой проект, включая подготовку проекта, индивидуальные консультации и защиту
- 36 академических часов самостоятельной работы на 1 курсовую работу, включая подготовку работы, индивидуальные консультации и защиту
- 18 академических часов самостоятельной работы на 1 расчетно-графическую работу, включая подготовку работы, индивидуальные консультации и защиту
- 9 академических часов самостоятельной работы на 1 индивидуальное домашнее задание, включая подготовку задания, индивидуальные консультации и защиту
- не менее 2 академических часов самостоятельной работы на консультации в период теоретического обучения и промежуточной аттестации

<sup>4</sup> включают предэкзаменационные консультации (при наличии), а также текущие консультации из расчета 10% от лекционных часов (приводятся к целому числу)

## 4. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

### 4.1 Наименование тем, их содержание и объем Курс 2 Семестр 2

№ п/п	Наименование раздела (краткое содержание)	Объем 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	16
2.	Принципы объектно-ориентированного программирования.				
	Абстрагирование, иерархическая организация, ограничение доступа, модульность. Определение объекта и класса. Объектная декомпозиция, диаграмма классов. Модули. Интерфейсы и реализации.	6	–	6	16
3.	Объектно-ориентированное проектирование.				
	Классы, виды отношений между классами. Наследование, полиморфизм и инкапсуляция. Классы в C++. Синтаксис и особенности C++.	8	–	6	14
4.	Дополнительные принципы объектно-ориентированного программирования				
	Многопоточность. Синхронизация потоков. Мьютексы и их реализация в библиотека STL. Устойчивость, области видимости и типы переменных. Типизация, проблемы приведения типов объектов одной иерархии.	4	–	2	14
5.	Проектирование компонент и модулей				
	Выделение внешних интерфейсов. Динамические ошибки. Исключительные ситуации. Проектирование и разработка структуры исключительных ситуаций.	4	–	4	18
6.	Создание шаблонов классов, стандартные шаблоны STL.				
	Шаблоны классов, механизм в C++ для его реализации. Изучение библиотек стандартных шаблонов (STL). Контейнеры объектов. Разработка пользовательских контейнеров.	4	–	8	14
7.	Шаблоны проектирования.				
	Порождающие шаблоны проектирования, структурные шаблоны проектирования, шаблоны поведения. Принципы SOLID. Тенденции и пути развития ООП.	4	–	4	15
	ВСЕГО:	34	–	34	107

## 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 консольного приложения в среде Microsoft Visual Studio	2	10
2	Принципы объектно-ориентированного программирования	Модульное программирование. Интерфейсы.	4	10
3	Введение в объектно-ориентированное программирование. Принципы объектно-ориентированного программирования.	Объектная декомпозиция	4	10
4	Принципы объектно-ориентированного программирования. Объектно-ориентированное проектирование.	Создание классов. Перегрузка операторов	4	10
5	Объектно-ориентированное проектирование.	Классы, виды отношений. Наследование.	4	12
6	Дополнительные принципы объектно-ориентированного программирования	Потоки в C++	2	12
7	Проектирование компонент и модулей	Обработка исключительных ситуаций в C++	4	10
8	Создание шаблонов классов, стандартные шаблоны STL.	Создание шаблонов классов в C++.	4	10
9	Создание шаблонов классов, стандартные шаблоны STL.	Использование стандартной библиотеки шаблонов STL	4	12
10	Шаблоны проектирования.	Умные указатели. Аллокаторы. Знакомство с QT	4	11
ИТОГО:			34	107

## 4.4. Содержание курсового проекта/работы<sup>5</sup>

<sup>5</sup> Если выполнение курсового проекта/курсовой работы нет в учебном плане, то в данном разделе необходимо указать «Не предусмотрено учебным планом»



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 записки (15-20 стр.), в которой отражаются все этапы создания программного продукта.

### **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 характеризуется 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 характеризуется 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 характеризуется 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<sup>6</sup>**

Учебным планом не предусмотрено

---

<sup>6</sup> Если выполнение расчетно-графического задания/индивидуального домашнего задания нет в учебном плане, то в данном разделе необходимо указать «Не предусмотрено учебным планом»

## 5. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ПРОВЕДЕНИЯ ТЕКУЩЕГО КОНТРОЛЯ, ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

### 5.1. Реализация компетенций

- 1. Компетенция ПКВ-1.** Способен осуществлять концептуальное, функциональное и логическое проектирование систем среднего и крупного масштаба и сложности.

Наименование индикатора достижения компетенции	Используемые средства оценивания
ПКВ-1.1. Знать: методы концептуального, функционального и логического проектирования систем среднего и крупного масштаба и сложности	защита лабораторной работы, работа над курсовой работой
ПКВ-1.2. Уметь: разрабатывать бизнес-требования к системе, определять цели создания системы, разрабатывать концепции системы	защита лабораторной работы, работа над курсовой работой
ПКВ-1.3. Владеть: навыками разработки технического задания на систему	защита лабораторной работы, экзамен, работа над курсовой работой

- 2. Компетенция ПКВ-3.** Способен разрабатывать требования и проектировать программное обеспечение

Наименование индикатора достижения компетенции	Используемые средства оценивания
ПКВ-3.1. Знать: виды требований к программному обеспечению и способы их документирования, методы и технологии проектирования программного обеспечения	защита лабораторной работы
ПКВ-3.2. Уметь: разрабатывать технические спецификации на программные компоненты и их взаимодействие	защита лабораторной работы, работа над курсовой работой
ПКВ-3.3. Владеть: навыками проектирования программного обеспечения	защита лабораторной работы, экзамен, работа над курсовой работой

## 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.	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.	Принципы объектно-ориентированного программирования	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.
3.	Объектно-ориентированное проектирование.	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.
4.	Дополнительные принципы объектно-ориентированного программирования	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
5.	Проектирование компонент и модулей	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.
6.	Создание шаблонов классов, стандартные шаблоны STL.	Библиотека стандартных классов (STL). Преобразование типов в C++. <code>dynamic_cast</code> , <code>static_cast</code> , <code>reinterpret_cast</code> , <code>const_cast</code> . Шаблоны классов, <code>template</code> . Контейнеры STL. Основные методы. Стандартная библиотека STL ( <code>map</code> , <code>list</code> ). Стандартная библиотека STL ( <code>vector</code> , <code>stack</code> ). Стандартная библиотека STL ( <code>set</code> , <code>queue</code> ).
7.	Шаблоны	Принципы SOLID (SO).

проектирования.	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.
-----------------	---

### 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.

Защита лабораторной работы проводится 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. Примерный перечень контрольных вопросов для защиты лабораторных работ приведен в таблице:

Тематика лабораторной работы	Контрольные вопросы
Лабораторная работа №1. Создание консольного приложения в среде Microsoft Visual Studio	1. Структура приложения в среде разработки MS Visual Studio 2013 2. Типы приложений, создаваемые в MS Visual Studio 2013. 3. Назначение и основные свойства стандартных компонентов Windows Forms 4. Что такое проект в MS Visual Studio 2013? Какие файлы входят в проект? 5. Отладка приложений в MS Visual Studio 2013.
Лабораторная работа №2. Модульное программирование. Интерфейсы.	1. Что такое модуль? 2. Какие бывают типы модулей в C++? 3. Структура данных типа «запись» (Struct). 4. Что такое интерфейс? 5. Причины сложности при разработке ПО.
Лабораторная работа №3. Объектная декомпозиция	1. Что такое объект? 2. Что такое объектная декомпозиция? 3. Отличия объектной декомпозиции от алгоритмической. 4. Как объекты взаимодействуют друг с другом? 5. Виды отношений между объектами.
Лабораторная работа №4.	1. Понятие класса.

Создание классов. Перегрузка операторов	2. Абстрагирование. 3. Принципы ООП. 4. Способы перегрузки операторов. 5. Простой полиморфизм.
Лабораторная работа №5. Классы, виды отношений. Наследование.	1. Полиморфизм. 2. Виды полиморфизма. 3. Виды отношения между классами 4. Виды методов. 5. Концептуальное отличие методы от функции.
Лабораторная работа №6. Потоки в C++	1. Дополнительные принципы ООП. 2. Виды параллельных вычислений. 3. Вычислительный поток. 4. Синхронизация потоков. 5. Особенности запуска потоков используя модуль <code>std::thread</code>
Лабораторная работа №7. Обработка исключительных ситуаций в C++	1. Работа с ошибками. 2. Проблемы при откладке ПО. 3. Исключительные ситуации. 4. Типы ошибок. 5. Иерархия исключительных ситуаций. 6. Принципы работы с исключениями в C++.
Лабораторная работа №8. Создание шаблонов классов в C++.	1. Шаблоны классов. 2. Отношение инстанцирования. 3. Виды аргументов при реализации шаблона в C++ 4. Шаблоны функций.
Лабораторная работа №9. Использование стандартной библиотеки шаблонов STL	1. Библиотека стандартных классов (STL). 2. Преобразование типов в C++. <code>dynamic_cast</code> , <code>static_cast</code> , <code>reinterpret_cast</code> , <code>const_cast</code> . 3. Шаблоны классов, <code>template</code> . 4. Контейнеры STL. Основные методы. 5. Стандартная библиотека STL ( <code>map</code> , <code>list</code> ). 6. Стандартная библиотека STL ( <code>vector</code> , <code>stack</code> ). 7. Стандартная библиотека STL ( <code>set</code> , <code>queue</code> ).
Лабораторная работа №10. Умные указатели. Аллокатеры. Знакомство с QT	1. Умные указатели. 2. Использование деструкторов. 3. Принципы работы с QTForm 4. Слоты и сигналы QT.

**Критерии оценки лабораторной работы:** лабораторная работа считается защищенной, если студент выполнил задание к работе полностью и во время устного опроса по работе правильно ответил на заданные преподавателем дополнительные вопросы.

#### 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<sup>7</sup>.

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

<sup>7</sup> В ходе текущей аттестации могут быть использованы балльно-рейтинговые шкалы.

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
	Владения 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	Не знает основные принципы и подходов, 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 рисунки и схемы 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 знания при декомпозиции задач предметной области	Не умеет использовать теоретические знания для выбора методики решения профессиональных задач	Использование теоретических знаний для выбора методики решения профессиональных задач вызывает затруднения	Умеет использовать теоретические знания для выбора методики решения профессиональных задач	Умело использует теоретические знания для выбора методики решения профессиональных задач

### Оценка сформированности компетенций по показателю Навыки.

Критерий	Уровень освоения и оценка			
	2	3	4	5
Владение навыков 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	Не может выполнить декомпозицию	Не достаточно качественно выполняет	Не достаточно качественно выполняет	Качественно выполняет оценку и выявления

требований к разработке программного продукта.	предметной области ни в каком виде.	исследования объектов предметной области, а также не качественно производит выявление требований к разрабатываемой системе	исследования объектов профессиональной деятельности, допускает и исправляет ошибки самостоятельно	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
4		

### 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 V6328633 Соглашение действительно с 02.10.2017 по 31.10.2020). Договор поставки ПО 0326100004117000038-0003147-01 от 06.10.2017.
2	Microsoft Office Professional Plus 2016	(Соглашение Microsoft Open Value Subscription V6328633 Соглашение действительно с 02.10.2017 по 31.10.2020). Договор поставки ПО 0326100004117000038-0003147-01 от 06.10.2017.
3	Kaspersky Endpoint Security «Стандартный Russian Edition».	Сублицензионный договор № 102 от 24.05.2018. Срок действия лицензии до 20.07.2019
4	Интегрированная среда разработки Microsoft Visual Studio 2013	Лицензионный договор № 63-14к от 2.07.2014
5	Среды программирования Free Pascal, Dev C++ или CodeBlocks	Свободно распространяемое ПО согласно условиям лицензионного соглашения

### 6.3. Перечень учебных изданий и учебно-методических материалов

Гарибов А. И. Объектно-ориентированное программирование : учеб. пособие для студентов бакалавриата 230100, 231000, 090303. – Белгород : Изд-во БГТУ им. В. Г. Шухова, 2015. – [Электронный ресурс]

2. Объектно-ориентированное программирование : методические указания к выполнению лабораторных работ и РГЗ / сост. А. И. Гарибов, Т. В. Бондаренко. – Белгород : Изд-во БГТУ им. В. Г. Шухова, 2015. – [Электронный ресурс]

3. Иванова Г. С. Объектно-ориентированное программирование : учебник / Г. С. Иванова, Т. Н. Ничушкина, Е. К. Пугачев. – Изд. 2-е, перераб. и доп. – М. : Изд-во МГТУ им. Н.Э. Баумана, 2003. – 367 с. – (Информатика в техническом университете). – ISBN 5-7038-2280-7

4. 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

5. Лаптев В. В. С++. Объектно-ориентированное программирование : учеб. пособие / В. В. Лаптев. - Санкт-Петербург : ПИТЕР, 2008. - 457 с
6. 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)

#### **17.4. Перечень интернет ресурсов, профессиональных баз данных, информационно-справочных систем**

1. Васильев А.Н. Объектно-ориентированное программирование на С++ [Электронный ресурс]/ А.Н. Васильев— Электрон. текстовые данные.— СПб.: Наука и Техника, 2016.— 544 с.— Режим доступа: <http://www.iprbookshop.ru/60648.html>.— ЭБС «IPRbooks».
2. Новиков П.В. Объектно-ориентированное программирование [Электронный ресурс]: учебно-методическое пособие к лабораторным работам/ П.В. Новиков— Электрон. текстовые данные.— Саратов: Вузовское образование, 2017.— 124 с.— Режим доступа: <http://www.iprbookshop.ru/64650.html>.— ЭБС «IPRbooks».
3. Мейер Б. Объектно-ориентированное программирование и программная инженерия [Электронный ресурс]/ Б. Мейер— Электрон. текстовые данные.— М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016.— 285 с.— Режим доступа: <http://www.iprbookshop.ru/39552.html>.— ЭБС «IPRbooks».
4. Самуйлов С.В. Объектно-ориентированное моделирование на основе UML [Электронный ресурс]: учебное пособие/ С.В. Самуйлов— Электрон. текстовые данные.— Саратов: Вузовское образование, 2016.— 37 с.— Режим доступа: <http://www.iprbookshop.ru/47277.html>.— ЭБС «IPRbooks».
5. Букунов С.В. Основы программирования на языке С++ [Электронный ресурс]: учебное пособие/ С.В. Букунов— Электрон. текстовые данные.— СПб.: Санкт-Петербургский государственный архитектурно-строительный университет, ЭБС АСВ, 2015.— 201 с.— Режим доступа: <http://www.iprbookshop.ru/63631.html>.— ЭБС «IPRbooks».
6. Комлев Н.Ю. Объектно Ориентированное Программирование. Хорошая книга для Хороших Людей. – М.: СОЛОН-ПРЕСС, 2015. – [Электронный ресурс]. – Режим доступа: <http://www.iprbookshop.ru/26923>.
7. С++ Standards Committee Papers [Электронный ресурс]. – Режим доступа: <http://www.open-std.org/jtc1/sc22/wg21/docs/papers/>
8. Документация по Visual C++. [Электронный ресурс]. – Режим доступа: <https://docs.microsoft.com/ru-ru/cpp/>
9. QT Creator [Электронный ресурс]. – Режим доступа: <https://www.qt.io/>
10. Code::Blocks [Электронный ресурс]. – Режим доступа: <http://www.codeblocks.org/>

## 7. УТВЕРЖДЕНИЕ РАБОЧЕЙ ПРОГРАММЫ<sup>3</sup>

Рабочая программа утверждена на 2020 /2021 учебный год  
без изменений / с изменениями, дополнениями<sup>4</sup>

Протокол № 8 заседания кафедры от «21» 04 2020 г.

Заведующий кафедрой \_\_\_\_\_ (Поляков В.М.)  
подпись, ФИО

Директор института \_\_\_\_\_ (Белоусов А.В.)  
подпись, ФИО

<sup>3</sup> Заполняется каждый учебный год на отдельных листах