

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования «Кабардино-Балкарский государственный  
университет им. Х.М. Бербекова» (КБГУ)

Институт информатики, электроники и робототехники  
Кафедра электроники и цифровых информационных технологий

СОГЛАСОВАНО

Руководитель образовательной про-  
граммы

 О.А. Молоканов

«16» декабря 2024 г.



Б.В. Шогенов

«16» декабря 2024 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)  
Б1.О.12.01 Объектно-ориентированное программирование

Специальность

12.05.01 Электронные и оптико-электронные приборы и системы специ-  
ального назначения

Специализация

Оптико-электронные информационно-измерительные приборы и системы

Квалификация (степень) выпускника

**Инженер**

Форма обучения

**Очная**

Нальчик 2024

Рабочая программа дисциплины (модуля) **«Объектно-ориентированное программирование»** /сост. О.Г.Ашхотов, И.Б.Ашхотова– Нальчик: КБГУ, 2024 г. 36 с.

Рабочая программа дисциплины (модуля) «Объектно-ориентированное программирование» предназначена для студентов очной формы обучения по специальности 12.05.01 Электронные и оптико-электронные приборы и системы специального назначения, 3 курс, 5,6 семестры.

Рабочая программа дисциплины (модуля) «Объектно-ориентированное программирование» составлена с учетом федерального государственного образовательного стандарта высшего образования по специальности **12.05.01 Электронные и оптико-электронные приборы и системы специального назначения**, утвержденного приказом Министерства образования и науки Российской Федерации «09» февраля 2018 г. № 93.

## Содержание

Содержание .....	3
1. Цель и задачи освоения дисциплины (модуля).....	4
2. Место дисциплины (модуля) в структуре ОПОП ВО .....	4
3. Требования к результатам освоения дисциплины (модуля).....	4
4. Содержание и структура дисциплины (модуля).....	4
Работа со строками .....	7
Работа с одномерным массивом.....	8
Работа с двумерным массивом.....	8
5. Оценочные материалы для текущего и рубежного контроля успеваемости и .....	8
5.1. Коллоквиум .....	8
5.2. Образцы тестовых заданий.....	11
5.4. Методические рекомендации по подготовке к тестированию .....	23
5.6. Задания для лабораторных занятий .....	24
6. Промежуточная аттестация .....	24
6.1. Методические рекомендации при подготовке к зачету .....	26
6.2. Список основных вопросов к устному экзамену.....	26
6.3. Распределение баллов текущего, рубежного контроля и экзамена.....	28
6.4. Критерии оценивания.....	28
7. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и опыта деятельности.....	28
8. Учебно-методическое обеспечение дисциплины (модуля).....	29
9. Программное обеспечение современных информационно-коммуникационных технологий.....	32
10. Материально-техническое обеспечение дисциплины .....	32
Приложение 1.....	35
Приложение 2.....	36

## 1. Цель и задачи освоения дисциплины (модуля)

**Целью дисциплины** «Объектно-ориентированное программирование» является приобретение базовых знаний и умений в соответствии с Федеральным государственным образовательным стандартом (ФГОС), формирование у студентов твердых теоретических и практических знаний в области разработки программного обеспечения на языках высокого уровня с использованием методов и средств объектно-ориентированного программирования, посредством обеспечения этапов формирования компетенций, предусмотренных ФГОС.

### **Основные задачи дисциплины:**

- изучение базовых концепций объектно-ориентированного программирования; освоение приемов решения задач профессиональной деятельности с использованием существующих информационно-коммуникационных технологий и с учетом основных требований информационной безопасности;
- освоение приемов, позволяющих сочетать при разработке программного обеспечения объектно-ориентированное и обобщенное программирование с использованием шаблонов;
- получение практических навыков разработки алгоритмических и программных решений в области системного и прикладного программного обеспечения.

Изучение дисциплины направлено на подготовку специалистов, способных решать проблемы, возникающие при эксплуатации изделий электронной техники с учетом области, типов и задач профессиональной деятельности в соответствии с профессиональными стандартами: профессиональный стандарт 29.004 "Специалист в области проектирования и сопровождения производства оптоэлектронных, оптических и оптико-электронных приборов и комплексов", утвержденный приказом Министерства труда и социальной защиты Российской Федерации от 24 декабря 2015 г. № 1141н.

## 2. Место дисциплины (модуля) в структуре ОПОП ВО

Дисциплина «Объектно-ориентированное программирование» включена в часть, формируемую участниками образовательных отношений Б1.О.12.01 учебного плана по специальности **12.05.01 Электронные и оптико-электронные приборы и системы специального назначения**, специализация: «Оптико-электронные информационно-измерительные приборы и системы».

Изучение дисциплины «Объектно-ориентированное программирование» базируется на следующих, ранее изучаемых, дисциплинах: «Цифровые и информационно-коммуникационные технологии», «Профильное программное обеспечение», «Алгоритмизация и программирование».

## 3. Требования к результатам освоения дисциплины (модуля)

Процесс изучения дисциплины направлен на формирование **обще профессиональных компетенций:**

**ОПК-4.** Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения.

**Код и наименование индикатора достижения компетенции:**

**ОПК-4.1.** Способен разрабатывать алгоритмы решения задач своей профессиональной деятельности.

**ОПК-4.2.** Способен разрабатывать программное обеспечение для решения задач своей профессиональной деятельности.

В результате изучения дисциплины (модуля) «Объектно-ориентированное программирование» студент должен:

**Знать** основы программирования: языки программирования (Python, Java, C++, и др.), принципы работы компьютера на низком уровне (процессор, память, устройства ввода/вывода).

**Уметь** работать с разными инструментами и средами разработки (IDE, системы контроля версий, библиотеки и т.д.)

**Владеть** навыками работы с базами данных и написание эффективных запросов.

## 4. Содержание и структура дисциплины (модуля)

В таблице 1 приводится описание содержания дисциплины, структурированное по разделам, с указанием по каждому разделу формы текущего контроля: защита лабораторной работы (ЛР), коллоквиум (К), тестирование (Т).

Таблица 1

	Наименование раздела	Содержание раздела/ темы	Код контролируемой компетенции (или ее части)	Наименование оценочного средства
1	Элементы языка	Алфавит языка. Комментарии. Типы данных. Константы-строки, или литералы. Директива препроцессора <code>define</code> . Описание. Модификатор <code>const</code> .	ОПК-4.1. ОПК-4.2.	К, Т, ЛР
2	Выражения	Операция и выражение присваивания. Арифметические операции. Операции отношения. Логические операции. Побитовые операции. Сдвиги. Операции автоувеличения и автоуменьшения <code>++</code> и <code>--</code> . Тернарная, или условная, операция. Операция следования. Приоритеты операций и порядок вычисления	ОПК-4.1. ОПК-4.2.	К, Т, ЛР
3	Указатели	Определение указателей. Указатели и массивы. Адресная арифметика. Символьные массивы и строки. Многомерные массивы. Указатели и многомерные массивы.	ОПК-4.1. ОПК-4.2.	К, Т, ЛР
4	Операция <code>sizeof</code> . Операции для работы с динамической памятью.	Операция выделения памяти <code>new</code> . Операция освобождения памяти <code>delete</code> .	ОПК-4.1. ОПК-4.2.	К, Т, ЛР
5	. Объявления и определения. Область существования имени	Компиляция, компоновка, библиотеки. Виды областей существования имени.	ОПК-4.1. ОПК-4.2.	К, Т, ЛР
6	Область видимости. Классы памяти. Объявления объектов и типов. Имена типов. Синоним имени типа. Правила преобразования стандартных типов.	Явные преобразования. Неявные преобразования стандартных базовых типов. Преобразование производных стандартных типов.	ОПК-4.1. ОПК-4.2.	К, Т, ЛР
7	Функции.	Определение и вызов функции. Передача аргументов в функцию. Передача многомерных массивов. Указатели на функции. Ссылки. Ссылки в качестве параметров функций. Рекурсивные функции. Аргументы по умолчанию. Перегрузка функций. Шаблоны функций.	ОПК-4.1. ОПК-4.2.	К, Т, ЛР

8	Перечисления. Классы.	Объявление классов. Конструкторы. Деструкторы. Статические члены класса. Указатель <i>this</i> . Статические функции-члены. Указатели на члены класса. Инициализация данных-членов класса. Конструктор копирования и операция присваивания. Дружественные функции. Конструктор и операция <i>new</i> . Вызов деструктора.	ОПК-4.1. ОПК-4.2.	<b>К, Т, ЛР</b>
9	Производные классы	Построение производного класса. Защищенные члены класса. Управление уровнем доступа к членам класса. Последовательность вызова конструктора и деструктора при построении производного класса на основе одного базового. Преобразования типов.	ОПК-4.1. ОПК-4.2.	<b>К, Т, ЛР</b>
10	Полиморфизм	Раннее и позднее связывание. Виртуальные функции. Абстрактные классы.	ОПК-4.1. ОПК-4.2.	<b>К, Т, ЛР</b>
11	Классы и шаблоны. Переопределение стандартных операций.	Основные определения и свойства. Операции <i>new</i> и <i>delete</i> при работе с абстрактными типами. Использование <i>new</i> при создании динамического объекта абстрактного типа. Операция <i>delete</i> . Преобразование типов.	ОПК-4.1. ОПК-4.2.	<b>К, Т, ЛР</b>
12	Некоторые особенности переопределенных операций	Операция = . Операция [ ]. Классы <i>Array</i> и <i>Matrix</i> .	ОПК-4.1. ОПК-4.2.	<b>К, Т, ЛР</b>
13	Списки	Операции над односвязными списками. Двухнаправленные и кольцевые списки. Операции над кольцевыми списками.	ОПК-4.1. ОПК-4.2.	<b>К, Т, ЛР</b>
14	Стеки	Реализация стека через массив. Реализация стека через динамическую цепочку звеньев.	ОПК-4.1. ОПК-4.2.	<b>К, Т, ЛР</b>
15	Двоичные деревья	Определение и построение. Таблицы.	ОПК-4.1. ОПК-4.2.	<b>К, Т, ЛР</b>
16	Потоковый ввод-вывод	Классы потоков. Стандартные потоки. Операции помещения и извлечения из потока. Форматирование потока. Файловый ввод-вывод с использованием потоков. Неформатируемый ввод-вывод. Часто применяемые функции. Файлы с произвольным доступом. Опрос и установка состояния потока. Переопределение операций извлечения и вставки. Переадресация ввода-вывода.	ОПК-4.1. ОПК-4.2.	<b>К, Т, ЛР</b>

### Структура дисциплины (модуля)

Таблица 2. Общая трудоемкость дисциплины составляет 8 зачетных единиц (288 часов).

Вид работы	Трудоемкость, часы		
	5 семестр	6 семестр	Всего
<b>Общая трудоемкость (в часах)</b>	<b>144</b>	<b>144</b>	<b>288</b>
<b>Контактная работа (в часах):</b>	<b>85</b>	<b>68</b>	<b>153</b>
<i>Лекционные занятия (Л)</i>	34	34	68

Лабораторные работы (ЛР)	51	34	85
<b>Самостоятельная работа (в часах), в том числе контактная работа:</b>	<b>50</b>	<b>49</b>	<b>99</b>
Курсовая работа (КР)/ Курсовой проект (КП)	-	3	3
Самостоятельное изучение разделов/тем	50	49	99
<b>Подготовка и прохождение промежуточной аттестации</b>	<b>9</b>	<b>27</b>	<b>36</b>
<b>Вид промежуточной аттестации</b>	<b>зачет</b>	<b>Экзамен, к/р</b>	<b>Зачет. Экзамен, к/р</b>

Таблица 3. Лекционные занятия  
5 семестр

№ п/п	Тема
1.	Предмет дисциплины и ее задачи.
2.	Элементы языка
3.	Выражения
4.	Указатели
5.	Операция sizeof. Операции для работы с динамической памятью.
6.	Объявления и определения. Область существования имени
7.	Область видимости. Классы памяти. Объявления объектов и типов. Имена типов. Синоним имени типа. Правила преобразования стандартных типов.
8.	Функции.
9.	Перечисления. Классы.

6 семестр

№ п/п	Тема
1.	Производные классы
2.	Полиморфизм
3.	Классы и шаблоны. Переопределение стандартных операций.
4.	Некоторые особенности переопределенных операций
5.	Списки
6.	Стеки
7.	Двоичные деревья
8.	Потоковый ввод-вывод

Таблица 4. Лабораторные работы  
5 семестр

№ п/п	Наименование лабораторных работ
1.	Программы с линейным алгоритмом
2.	Целочисленные типы. Использование функций
3.	Работа с данными
4.	Запись числа с плавающей точкой
5.	Работа со строками

6.	Строчно-ориентированный ввод с помощью getline() или get(). Смешивание строкового и числового ввода.
7.	Массивы в C++. Объявление, инициализация, использование
8.	Работа с одномерным массивом.
9.	Работа с двумерным массивом.
10.	Введение в класс string
11.	Указатели. Инициализация указателей. Выделение памяти с помощью операции new
12.	Введение в циклы for
13.	Цикл while. Цикл do while
14.	Использование функций.

### 6 семестр

№	Наименование лабораторных работ
1.	Указатели
2.	Подпрограммы. Указатели.
3.	Подпрограммы. Функции
4.	Рекурсивные функции
5.	Передача параметра по ссылке
6.	Динамическое выделение памяти
7.	Особенности мультифайлового программирования
8.	Массивы. Связь указателей и массивов.
9.	Определение максимального элемента и его положения в массиве.
10.	Классы и объекты.
11.	Конструкторы и деструкторы
12.	Инкапсуляция, наследование, полиморфизм
13.	Работа с текстом, файлами, строками
14.	Создание интерфейса пользователя в текстовом режиме
15.	Интерполирование алгебраическими многочленами

Таблица 6. Самостоятельное изучение разделов дисциплины

№ п/п	Вопросы, выносимые на самостоятельное изучение
1.	Перегрузка операций, друзья класса
2.	Наследование классов
3.	Обобщенное программирование, шаблоны
4.	Обработка исключительных ситуаций
5.	Принципы организации стандартной библиотеки языка C++
6.	Последовательные контейнеры
7.	Ассоциативные контейнеры
8.	Адаптеры последовательных контейнеров
9.	Итераторы, функциональные объекты
10.	Алгоритмы стандартной библиотеки

## 5. Оценочные материалы для текущего и рубежного контроля успеваемости и промежуточной аттестации

### 5.1. Коллоквиум

В 5 и 6 семестрах проводятся по 3 коллоквиума, которые оцениваются по 8 баллов каждый.

#### 5.1.1. Вопросы, выносимые на коллоквиум (контролируемые компетенции ОПК-4.1, ОПК-4.2)

### 5 семестр



### Первый коллоквиум

1. Алфавит языка.
2. Комментарии.
3. Типы данных.
4. Константы-строки, или литералы.
5. Директива препроцессора `define`.
6. Описание. Модификатор `const`.
7. Операция и выражение присваивания.
8. Арифметические операции.
9. Операции отношения.
10. Логические операции.
11. Побитовые операции.
12. Сдвиги.
13. Операции автоувеличения и автоуменьшения `++` и `--`.
14. Тернарная, или условная, операция.
15. Операция следования.
16. Приоритеты операций и порядок вычисления

### Второй коллоквиум

1. Определение указателей.
2. Указатели и массивы.
3. Адресная арифметика.
4. Символьные массивы и строки.
5. Многомерные массивы.
6. Указатели и многомерные массивы.
7. Операция выделения памяти `new`.
8. Операция освобождения памяти `delete`.
9. Компиляция, компоновка, библиотеки.
10. Виды областей существования имени.

### Третий коллоквиум

1. Явные преобразования.
2. Неявные преобразования стандартных базовых типов.
3. Преобразование производных стандартных типов.
4. Определение и вызов функции.
5. Передача аргументов в функцию.
6. Передача многомерных массивов.
7. Указатели на функции.
8. Ссылки.
9. Ссылки в качестве параметров функций.
10. Рекурсивные функции.
11. Аргументы по умолчанию.
12. Перегрузка функций.
13. Шаблоны функций.
14. Объявление классов.
15. Конструкторы. Деструкторы.
16. Статические члены класса.
17. Указатель `this`. Статические функции-члены.
18. Указатели на члены класса. Инициализация данных-членов класса.
19. Конструктор копирования и операция присваивания.
20. Дружественные функции. Конструктор и операция `new`. Вызов деструктора.

### 6 семестр Первый коллоквиум

1. Построение производного класса.

2. Защищенные члены класса.
3. Управление уровнем доступа к членам класса.
4. Последовательность вызова конструктора и деструктора при построении производного класса на основе одного базового.
5. Преобразования типов.
6. Раннее и позднее связывание.
7. Виртуальные функции.
8. Абстрактные классы.
9. Основные определения и свойства класса и шаблона.
10. Операции *new* и *delete* при работе с абстрактными типами.
11. Использование *new* при создании динамического объекта абстрактного типа.
12. Операция *delete*.
13. Преобразование типов.

### **Второй коллоквиум**

1. Особенности переопределенных операций
2. Операция  $=$ . Операция  $[]$ . Классы `Array` и `Matrix`.
3. Операции над односвязными списками.
4. Двухнаправленные и кольцевые списки.
5. Операции над кольцевыми списками.
6. Реализация стека через массив.
7. Реализация стека через динамическую цепочку звеньев.
8. Определение и построение. Таблицы.

### **Третий коллоквиум**

1. Классы потоков.
2. Стандартные потоки.
3. Операции помещения и извлечения из потока.
4. Форматирование потока.
5. Файловый ввод-вывод с использованием потоков.
6. Неформатируемый ввод-вывод.
7. Часто применяемые функции.
8. Файлы с произвольным доступом.
9. Опрос и установка состояния потока.
10. Переопределение операций извлечения и вставки.
11. Переадресация ввода-вывода.

#### ***Рекомендации при подготовке к коллоквиуму***

- проработать конспекты лекций по вопросам коллоквиума;
- прочитать основную и дополнительную литературу, рекомендованную по изучаемым вопросам;
- ответить на вопросы коллоквиума;
- при затруднениях, проконсультироваться с преподавателем.

#### ***Критерии оценивания***

<b>Оценка</b>			
<b>Неудовлетворительно 2 балла</b>	<b>Удовлетворительно 4 балла</b>	<b>хорошо 6 баллов</b>	<b>отлично 8 баллов</b>

Студент не знает значительной части вопросов, допускает существенные ошибки в ответах на вопросы	Студент поверхностно знает вопросы коллоквиума, допускает неточности в ответе на вопрос	Студент хорошо знает материал, Грамотно и по существу излагает его, допуская некоторые неточности в ответе на вопрос	Студент в полном объеме знает материал, грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопрос.
--	---	--	---

**5.2. Образцы тестовых заданий**  
(контролируемые компетенции ОПК-4.1, ОПК-4.2)

**5 семестр**

**1-я контрольная точка**

I:

S: Типы bool, int, char являются ...

- фундаментальными типами
- перечислениями
- структурами
- типами, определяемыми пользователем

I:

S: Типы float, double являются ...

- интегральными типами
- типами с плавающей точкой
- классами
- типами, определяемыми пользователем

I:

S: Типы bool, int, char являются ...

- интегральными типами
- типами с плавающей точкой
- объединениями
- типами, определяемыми пользователем

I:

S: Классы и структуры являются ...

- арифметическими типами
- интегральными типами
- символьными типами
- типами, определяемыми пользователем

I:

S: Типы double, int, char являются ...

- типами с плавающей точкой
- символьными типами
- арифметическими типами
- интегральными типами

I:

S: Перечисления и объединения являются ...

- арифметическими типами
- интегральными типами
- символьными типами
- типами, определяемыми пользователем

I:

S: Создайте переменную с произвольным именем, являющуюся указателем на константные данные типа с плавающей точкой.

I:

S: Создайте переменную целого типа с произвольным именем.

Создайте переменную, являющуюся константной ссылкой на первую переменную.

I:  
 S: Создайте переменную с произвольным именем, являющуюся статическим массивом символов (размер произвольный).  
 I:  
 S: Создайте переменную логического типа с произвольным именем. Создайте вторую переменную, хранящую адрес первой переменной.  
 I:  
 S: Создайте переменную с произвольным именем и интегральным типом. Создайте вторую переменную, являющуюся ссылкой на первую переменную.  
 I:  
 S: Создайте переменную с произвольным именем, являющуюся статическим двумерным массивом целых чисел (размеры произвольные).  
 I:  
 S: Создайте переменную с произвольным именем, являющуюся константным указателем на константные данные целого типа.  
 I:  
 S: Создайте переменную логического типа с произвольным именем. Создайте вторую переменную, являющуюся ссылкой на первую переменную.

## 2-я контрольная точка

I:  
 S: Оператор, позволяющий узнать размер типа на используемой платформе:  
 - typedef  
 - sizeof  
 - тернарный оператор  
 - бинарный оператор  
 I:  
 S: Согласно стандарту выполняется следующее соотношение:  
 - размер(char) = размер(short) = размер(int) = размер(long)  
 - размер(char) ≤ размер(short) ≤ размер(int) ≤ размер(long)  
 - размер(char) ≥ размер(short) ≥ размер(int) ≥ размер(long)  
 I:  
 S: Согласно стандарту выполняется следующее соотношение:  
 - размер(float) = размер(double) = размер(long double)  
 - размер(float) ≤ размер(double) ≤ размер(long double)  
 - размер(float) ≥ размер(double) ≥ размер(long double)  
 I:  
 S: Все размеры в C++ кратны размеру типа ...  
 - int  
 - char  
 - float  
 - short  
 I:  
 S: Объявите функцию без параметров, возвращающую указатель на данные целого типа. Имя функции произвольное.  
 I:  
 S: Объявите функцию с одним аргументом типа string, передаваемым по ссылке. Возвращаемый тип – логический. Имя функции произвольное.  
 I:  
 S: Объявите не возвращающую ничего функцию с двумя аргументами целого типа, передаваемыми по значению. Имя функции произвольное.  
 I:  
 S: Объявите функцию без параметров, возвращающую константную ссылку на значение типа с плавающей точкой. Имя функции произвольное.  
 I:  
 S: Объявите функцию с одним аргументом типа double, передаваемым через указатель. Возвращаемый тип – целое число. Имя функции произвольное.

I:

S: Объявите не возвращающую ничего функцию с двумя аргументами целого типа, передаваемыми по константной ссылке. Имя функции произвольное.

I:

S: Какой результат будет выведен в консоль при выполнении функции main?

```
void main() {  
    int i=10;  
    for (i=i+1; i<12; ++i)  
        cout << i;
```

- }  
-     ничего
- 10
- 11
- 1011

I:

S: Какой результат будет выведен в консоль при выполнении функции main?

```
void main() {  
    int i = 10;  
    if (i = 11)  
        cout << i;
```

- }  
-     ничего
- 10
- 11
- 0

I:

S: Какой результат будет выведен в консоль при выполнении функции main?

```
void main() {  
    int i=10;  
    for (i=i+1; i<11; ++i)  
        cout << i;
```

- }  
-     ничего
- 10
- 11
- 1011

I:

S: Какой результат будет выведен в консоль при выполнении функции main?

```
void main() {  
    int i=10;  
    if (i += 1)  
        cout << i;
```

- }  
-     ничего
- 10
- 11
- 0

I:

S: Какой результат будет выведен в консоль при выполнении функции main?

```
void main() {  
    int i=11;  
    for (i=i-1; i<12; ++i)  
        cout << i;
```

- }  
-     ничего
- 10
- 11

- 1011

### 3-я контрольная точка

I:

S: Какой результат будет выведен в консоль при выполнении функции main?

```
void calc(int x)
{ x = x+5; }
```

```
void main() {
    int x=5;
    calc(x);
    cout << x;
}
```

- точно определить невозможно

- 0

- 5

- 10

I:

S: Какой результат будет выведен в консоль при выполнении функции main?

```
void calc(int* p)
{ *p -= 5; }
```

```
void main() {
    int x=10;
    calc(&x);
    cout << x-5;
}
```

- точно определить невозможно

- 0

- 5

- 10

I:

S: Какой результат будет выведен в консоль при выполнении функции main?

```
int calc(int& x)
{ return x+5; }
```

```
void main() {
    int x=0;
    calc(x);
    cout << calc(x);
}
```

- точно определить невозможно

- 0

- 5

- 10

I:

S: Какой результат будет выведен в консоль при выполнении функции main?

```
int calc(int x)
{ return x-5; }
```

```
void main() {
    int x=10;
    x = calc(5);
    cout << x+5;
}
```

- точно определить невозможно

- 0

- 5
- 10

I:

S: Какой результат будет выведен в консоль при выполнении функции main?

```
void calc(int* p)
{ p = 0; }
```

```
void main() {
    int x=10;
    calc(&x);
    cout << x;
}
```

- точно определить невозможно

- 0
- 5

-

I:

S: Чему будет равна переменная X?

```
int X(6);
X -= X + 3 * 2;
```

- -6
- 18
- 12
- 6

I:

S: Чему будет равна переменная X?

```
double Y(5.2);
int X = (Y > 5) ? 6 : 5;
```

- 5
- 5.2
- 6
- 6.2

I:

S: Чему будет равна переменная X?

```
int Y(5);
int X = Y - 3 * 2;
```

- 4
- 0
- -1
- 1

I:

S: Чему будет равна переменная X?

```
bool Y(true);
int X = (!Y) ? 3+Y : 5+Y;
```

- 3
- 4
- 5
- 6

I:

S: Чему будет равна переменная X?

```
double X(5.5);
X += 5.5-1;
```

- 5.5
- 11
- 10
- 4.5

**6 семестр**  
**1-я контрольная точка**

I:

S: В приведенном фрагменте кода происходит ...

```
string *pt = new string;
```

```
delete pt;
```

- выделение и освобождение памяти под объект типа string
- выделение и некорректное освобождение памяти под объект типа string
- выделение и освобождение памяти под массив объектов типа string
- выделение и некорректное освобождение памяти под массив объектов типа string

I:

S: В приведенном фрагменте кода происходит ...

```
double *Ptr = new double;
```

```
delete[] Ptr;
```

- выделение и освобождение памяти под значение типа double
- выделение и некорректное освобождение памяти под значение типа double
- выделение и освобождение памяти под массив значений типа double
- выделение и некорректное освобождение памяти под массив значений типа double

I:

S: В приведенном фрагменте кода происходит ...

```
string *pt = new string[7];
```

```
delete[] pt;
```

- выделение и освобождение памяти под объект типа string
- выделение и некорректное освобождение памяти под объект типа string
- выделение и освобождение памяти под массив объектов типа string
- выделение и некорректное освобождение памяти под массив объектов типа string

I:

S: В приведенном фрагменте кода происходит ...

```
double *Ptr = new double[10];
```

```
delete Ptr;
```

- выделение и освобождение памяти под значение типа double
- выделение и некорректное освобождение памяти под значение типа double
- выделение и освобождение памяти под массив значений типа double
- выделение и некорректное освобождение памяти под массив значений типа double

I:

S: Метод класса, который вызывается автоматически при создании объекта, называется ...

- дружественным
- конструктором
- константным
- деструктором

I:

S: Метод класса, который вызывается автоматически при уничтожении объекта, называется

- дружественным
- конструктором
- константным
- деструктором

I:

S: Конструктор класса может быть ...

- константным методом
- виртуальным методом
- и константным, и виртуальным
- никаким из перечисленных

I:

S: Деструктор класса может быть ...

- константным методом
- виртуальным методом
- и константным, и виртуальным



- никаким из перечисленных

I:

S: Сколько аргументов может принимать деструктор класса?

-  $\geq 0$

- всегда 1

-  $> 0$

- всегда 0

I:

S: Что является возвращаемым типом конструктора класса?

- тип первого поля класса

- тип, указанный при объявлении конструктора

- сам класс

- конструктор ничего не возвращает

I:

S: Сколько аргументов может принимать конструктор класса?

-  $\geq 0$

- всегда 1

- всегда 0

I:

S: Что является возвращаемым типом деструктора класса?

- тип первого поля класса

- тип, указанный при объявлении деструктора

- сам класс

- деструктор ничего не возвращает

## 2-я контрольная точка

I:

S: Какой оператор используется для доступа к открытому содержимому класса при работе с объектом класса?

- .

- ->

- \*

- никакой из перечисленных

I:

S: Какой оператор используется для доступа к закрытому содержимому класса при работе с указателем на объект класса?

- .

- ->

- \*

- никакой из перечисленных

I:

S: Какой оператор используется для доступа к закрытому содержимому класса при работе с объектом класса?

- .

- ->

- \*

- никакой из перечисленных

I:

S: Какой оператор используется для доступа к открытому содержимому класса при работе с указателем на объект класса?

- .

- ->

- \*

- никакой из перечисленных

I:

S: Защищенные поля и методы класса находятся после спецификатора ...

- public

- friend
- private
- protected

I:

S: Спецификатор private разрешает доступ к содержимому класса ...

- только из методов этого класса
- из методов этого класса и методов его наследников
- из любого места программы

I:

S: По умолчанию в классе используется спецификатор доступа ...

- public
- protected
- private
- спецификатор доступа не определён

I:

S: Спецификатор public разрешает доступ к содержимому класса ...

- только из методов этого класса
- из методов этого класса и методов его наследников
- из любого места программы

I:

S: Закрытые поля и методы класса находятся после спецификатора ...

- public
- friend
- private
- protected

I:

S: Спецификатор protected разрешает доступ к содержимому класса ...

- только из методов этого класса
- из методов этого класса и методов его наследников
- из любого места программы

I:

S: Открытые поля и методы класса находятся после спецификатора ...

- public
- friend
- private
- protected

I:

S: Как будет выглядеть объявление конструктора без параметров для приведенного класса?

```
class Car {
    string name;
public:
    string getName() const;
};
- void Car();
- Constructor();
- Car(string s);
- Car();
- string Car();
```

I:

S: Как будет выглядеть объявление деструктора для приведенного класса?

```
class Cat {
    string name;
public:
    string getName() const;
};
- void Cat();
- string ~Cat();
```

- Destructor();
- ~Cat();
- ~Cat(string s);

I:

S: Как будет выглядеть объявление конструктора без параметров для приведенного класса?

```
class Human {
    string name;
public:
    string getName() const;
};
```

- Constructor();
- Constructor(string s);
- ~Human();
- Human();
- void Human();

I:

S: Как будет выглядеть объявление деструктора для приведенного класса?

```
class House {
    string name;
public:
    string getName() const;
};
```

- Destructor();
- ~House();
- ~Destructor();
- void ~House();
- string ~House();

I:

S: Как будет выглядеть объявление конструктора с параметрами для приведенного класса?

```
class Base {
    string name;
public:
    string getName() const;
};
```

- Base();
- ~Base();
- Base(string s);
- ~Base(string s);
- void Base();

I:

S: Как будет выглядеть объявление конструктора без параметров для приведенного класса?

```
class Boss {
    string name;
public:
    string getName() const;
};
```

- Boss();
- ~Boss();
- Boss(string s);
- ~Boss(string s);
- void Boss();

I:

S: Дружественным функциям разрешен доступ ...

- ко всему содержимому класса
- ко всему содержимому класса и содержимому классов-наследников
- к открытому и защищенному содержимому класса
- к открытому содержимому класса

- все перечисленные варианты не верны
- I:
- S: Вложенному (объявленному в другом классе) классу разрешен доступ ...
- ко всему содержимому основного класса
  - ко всему содержимому основного класса и содержимому его наследников
  - к открытому и защищенному содержимому основного класса
  - к открытому содержимому основного класса
  - все перечисленные варианты не верны
- I:
- S: Методы класса могут обращаться ...
- ко всему содержимому класса
  - к открытому и защищенному содержимому класса
  - к открытому содержимому класса
  - все перечисленные варианты не верны
- I:
- S: Методы класса-наследника могут обращаться ...
- ко всему содержимому базового класса
  - к открытому и защищенному содержимому базового класса
  - к открытому содержимому базового класса
  - все перечисленные варианты не верны

### 3-я контрольная точка

- I:
- S: Внутри константных методов класса можно ...
- изменять значения полей
  - получать значения полей
  - изменять и получать значения полей
  - ничего из перечисленного
- I:
- S: Внутри константных методов класса можно ...
- вызывать константные методы класса
  - вызывать неконстантные методы класса
  - вызывать константные и неконстантные методы класса
  - ничего из перечисленного
- I:
- S: Внутри константных методов класса можно ...
- получать значения полей и вызывать неконстантные методы класса
  - изменять значения полей и вызывать любые методы класса
  - получать значения полей и вызывать константные методы класса
  - ничего из перечисленного
- I:
- S: В функции func происходит ...
- ```
class Student {
    string Name;
public:
    void setName(string s)
    { Name = s; }
    string getName() const
    { return Name; }
};
void func(Student &A) {
    A.setName("Ivanov");
}
```
- изменение состояния объекта A
  - ошибка при компиляции
  - получение состояния объекта A
  - точно определить невозможно

I:  
S: В функции func происходит ...  
class Calc {  
 int val;  
public:  
 void setVal(string s);  
 int getResult();  
};  
void func(const Calc& A) {  
 int r = A.getResult();  
}  
- изменение состояния объекта A  
- ошибка при компиляции  
- получение состояния объекта A  
- точно определить невозможно

I:  
S: В функции func происходит ...  
class Student {  
 string Name;  
public:  
 void setName(string s)  
 { Name = s; }  
 string getName() const  
 { return Name; }  
};  
void func(Student &A) {  
 string s = A.getName();  
}  
- изменение состояния объекта A  
- ошибка при компиляции  
- получение состояния объекта A  
- точно определить невозможно

I:  
S: В функции func происходит ...  
class Calc {  
 int val;  
public:  
 void setVal(string s);  
 int getResult();  
};  
void func(Calc &A) {  
 int r = A.getResult();  
}  
- изменение состояния объекта A  
- получение состояния объекта A  
- точно определить невозможно

I:  
S: В функции func происходит ...  
class Student {  
 string Name;  
public:  
 void setName(string s)  
 { Name = s; }  
 string getName() const  
 { return Name; }  
};  
void func(const Student &A) {

```

A.setName("Ivanov");
}
- изменение состояния объекта A
- ошибка при компиляции
- получение состояния объекта A
- точно определить невозможно
I:
S: В функции func состояние объекта A ...
class Calc {
    int val;
public:
    void setVal(string s);
    int getResult() const;
};
void func(Calc &A) {
    int r = A.getResult();
}
- изменяется
- не изменяется
- точно определить невозможно

```

I:  
S: В какой строке приведенного фрагмента кода вызывается конструктор копии класса Student?

```

1 Student A;
2 Student B("Ivanov");
3 Student C(B);
4 A = Student("Petrov");
- 1
- 2
- 3
- 4

```

I:  
S: Как будет выглядеть объявление конструктора копии для класса Comp?

```

- Comp(const Comp& x);
- Copy(const Comp& x);
- Comp& Comp();
- Comp Comp(const Comp& x);

```

I:  
S: Какая строка будет использоваться вместо многоточия для защиты от самоприсваивания?  
Calc& Calc::operator=(const Calc& a) {

```

...
val = a.val;
return *this;
}
- if (this == &a)
- if (this != &a)
- if (this == a)
- if (this != a)

```

I:  
S: В какой строке приведенного фрагмента кода вызывается оператор присваивания класса Student?

```

1 Student X("Ivanov");
2 string s = X.getName();
3 Student Y(X);
4 Y = Student("Petrov");
- 1
- 2
- 3

```

- 4

I:

S: Какая строка будет использоваться вместо многоточия для защиты от самоприсваивания?

```
Cat& Cat::operator=(const Cat& c) {
```

```
...
```

```
name = c.name;
```

```
return *this;
```

```
}
```

- if (\*this != &c)

- if (\*this == &c)

- if (this != &c)

- if (this == &c)

I:

S: В какой строке приведенного фрагмента кода вызывается конструктор копии класса Car?

```
void func1(Car z) { }
```

```
void func2(Car &z) { }
```

```
...
```

```
1 Car A("Car");
```

```
2 Car B;
```

```
3 func1(A);
```

```
4 B = A;
```

```
5 func2(B);
```

- 1

- 2

- 3

- 4

- 5

#### 5.4. Методические рекомендации по подготовке к тестированию

Тесты – это вопросы или задания, предусматривающие конкретный, краткий, четкий ответ на имеющиеся эталоны ответов. При самостоятельной подготовке к тестированию студенту необходимо:

- готовясь к тестированию, проработать информационный материал по дисциплине. Проконсультироваться с преподавателем по вопросу выбора учебной литературы;
- четко выясните все условия тестирования заранее. Знать, сколько тестов Вам будет предложено, сколько времени отводится на тестирование, какова система оценки результатов и т.д.
- приступая к работе с тестами, внимательно и до конца прочтите вопрос и предлагаемые варианты ответов. Выберите правильные (их может быть несколько). На отдельном листке ответов выпишите цифру вопроса и буквы, соответствующие правильным ответам;
- в процессе решения желательно применять несколько подходов в решении задания. Это позволяет максимально гибко оперировать методами решения, находя каждый раз оптимальный вариант.
- если Вы встретили чрезвычайно трудный для Вас вопрос, не тратьте много времени на него. Переходите к другим тестам. Вернитесь к трудному вопросу в конце.
- обязательно оставьте время для проверки ответов, чтобы избежать механических ошибок.

#### 5.5. Критерии оценивания

| Оценка                          |                              |                   |                     |
|---------------------------------|------------------------------|-------------------|---------------------|
| неудовлетворительно<br>0 баллов | удовлетворительно 3<br>балла | хорошо<br>4 балла | отлично<br>5 баллов |

|                                           |                                       |                                       |                                        |
|-------------------------------------------|---------------------------------------|---------------------------------------|----------------------------------------|
| Менее 50 % правильно выполненных заданий. | 50-70% правильно выполненных заданий. | 71-85% правильно выполненных заданий. | 86-100% правильно выполненных заданий. |
|-------------------------------------------|---------------------------------------|---------------------------------------|----------------------------------------|

### **5.6. Задания для лабораторных занятий** (контролируемые компетенции ОПК-4.1, ОПК-4.2)

Лабораторный практикум является важным элементом обучения, т.к. прививает навыки самостоятельной работы на различном лабораторном оборудовании и умение пользоваться различными приборами и инструментами.

#### **Пример типовой лабораторной работы «Программы с линейным алгоритмом»**

**Цель работы:** овладение практическими навыками программирования линейных алгоритмических конструкций, а также работы с компилятором С/С++.

Даются 6 готовых программ с подробным разбором каждой операции. Далее необходимо самостоятельно составить 5 задач с проверкой на работоспособность.

#### **Методические рекомендации**

Выполнение каждой лабораторной работы складывается из следующих этапов.

1. Самостоятельная подготовка студентов к работе. Перед началом работы студенты должны четко представлять себе цель работы, знать основы языка программирования. Студенты, не подготовившиеся к работе в соответствии с этими требованиями, к выполнению работы не допускаются.

2. Проведение эксперимента. Этот этап осуществляется в соответствии с методическими указаниями, которые содержатся в описании к каждой работе. Лабораторные измерения на стенде студент может начать только после собеседования с преподавателем и получения соответствующего допуска. Результаты лабораторной работы проверяются преподавателем.

При работе в лаборатории необходимо строго выполнять все правила техники безопасности и указания преподавателя.

3. Составление отчета о проделанной работе. К отчету о выполненной работе предъявляются следующие требования:

Отчет должен содержать исчерпывающие данные, как о цели работы, так и о результатах в следующей последовательности:

- задание;
- схема установки и описание методики измерений;
- первичные экспериментальные результаты за подписью преподавателя;
- результаты обработки экспериментальных данных, включая графики, таблицы;
- общие выводы о работе и заключение, о качестве исследованных материалов.

Текст отчета должен быть написан аккуратно и разборчиво от руки или представлен в виде распечатки, после компьютерной верстки. В обоих случаях текст должен представлять собой логическое изложение существа вопроса. Недопустимо приведение формул, таблиц без разъяснений всех обозначений и сокращений. Отчет должен быть понятен для каждого читающего без каких-либо дополнительных вопросов у составителей отчета.

4. После представления отчета студент должен иметь, как минимум, поверхностные знания по контрольным вопросам к работе, имеющимся в методических указаниях, и ему выставляется балл, которым оценена данная лабораторная работа.

### **6. Промежуточная аттестация** (контролируемые компетенции ОПК-4.1, ОПК-4.2)

#### **Темы курсовых работ**

1. Реализация контейнера на основе комбинированной структуры «Стек-массив динамических списков»
2. Разработка объектной программы для задачи «Строительная организация»
3. Объектная реализация контейнера на основе комбинированной структуры «Двунаправленный список массивов-стеков»



4. Объектная реализация контейнера на основе комбинированной структуры «Упорядоченный массив динамических очередей»
5. Разработка объектной программы для задачи учета компьютерной техники в организации
6. Двухнаправленный список
7. Массив динамических очередей
8. Объектная реализация контейнера на основе комбинированной структуры «Динамический список массивов-стеков»
9. Объектная реализация контейнера на основе комбинированной структуры «Динамический список массивов-стеков»
10. Объектная реализация контейнера на основе комбинированной структуры «Стек-массив динамических списков»
11. Объектная реализация контейнера на основе комбинированной структуры «Очередь-массив двухнаправленных списков»
12. База данных
13. Разработка набора взаимосвязанных классов для реализации заданной сложной структуры контейнера
14. Разработка набора взаимосвязанных классов для реализации заданной сложной структуры данных как специализированного контейнера
15. Разработка DLL библиотеки по работе с графами
16. Разработка криптографического программного обеспечения «Шифровка и дешифровка текста»
17. Основы работы в интегрированной среде разработки Delphi, основные компоненты и их свойства
18. Основы работы в интегрированной среде программирования Delphi
19. Выходной тестовый контроль
20. Автоматизированная обучающая система

### **6.1. Вопросы к теоретическому зачету 5 семестр**

- 1) Основные понятия и сферы применения объектно-ориентированного программирования. Преимущества объектно-ориентированного подхода по отношению к процедурному подходу к разработке программ.
- 2) Предпосылки и история создания языка C++. Достоинства и недостатки языка C++. Сферы применения языка C++.
- 3) Компиляторы языка C++. Средства разработки программ на языке C++.
- 4) Классификация типов данных в языке C++. Целые типы данных. Символьные типы данных. Типы для представления чисел с плавающей точкой. Логический тип данных. Тип void.
- 5) Оператор sizeof(). Гарантии стандарта по соотношению размеров встроенных типов. Получение информации о встроенных типах данных для конкретной платформы.
- 6) Классификация типов данных в языке C++. Литералы встроенных типов.
- 7) Структура и особенности объявления имен в языке C++. Имена (идентификаторы). Область видимости. Вложенные области видимости.
- 8) Пространства имен. Создание псевдонимов для типов данных, ключевое слово typedef.
- 9) Инициализация сущностей в языке C++. Имена (идентификаторы). Область видимости.
- 10) Правила инициализации переменных в разных областях видимости при отсутствии инициализатора.
- 11) Стандартные операции в языке C++. Арифметические операции. Преобразование типов в арифметических выражениях. Приоритет операций.
- 12) Логические операции. Побитовые логические операции. Операции присваивания. Операция запятой. Приоритет операций.
- 13) Управляющие конструкции в языке C++. Оператор ветвления if. Тернарная операция. Оператор выбора вариантов switch.
- 14) Операторы циклов while и do-while. Оператор цикла for. Операторы прерывания циклов break и continue.
- 15) Определение и основы работы с указателями и массивами в языке C++. Многомерные массивы. Связь между указателями и массивами, индексация с помощью указателей.

- 16) Константы. Константные указатели и указатели на константу. Ссылки. Константные ссылки.
- 17) Перечисления. Структуры. Битовые поля. Объединения. Опережающие объявления. Эквивалентность типов.
- 18) Объявление и определение функции в языке C++. Передача параметров в функцию по значению. Возврат результатов работы функции.
- 19) Передача параметров по ссылке и константной ссылке. Передача параметров в функцию через указатель. Возврат результатов работы функции.
- 20) Передача массивов в качестве параметров функции. Многомерные массивы в качестве аргументов функции.
- 21) Использование внутри функции переменной из охватываемой области видимости. Статические переменные в функциях. Встраиваемые (inline) функции. Перегрузка функций.
- 22) Перегрузка функций. Аргументы по умолчанию. Функции с переменным числом аргументов. Указатели на функции.
- 23) Общие сведения о приведении типов. Приведение типов в стиле языка C. Приведение типов в функциональном стиле. Операторы приведения типов в языке C++.
- 24) Структура памяти программы. Модель памяти с точки зрения программиста на языке C++. Создание динамических объектов (размещение объектов в куче).
- 25) Динамические массивы. Многомерные динамические массивы.
- 26) Организация многофайловых программ в языке C++. Спецификатор extern. Директива #include. Защита от множественного включения заголовочных файлов.
- 27) Понятие о препроцессоре, директивы препроцессора. Директива #define, макросы. Макросы с параметрами. Директивы условной компиляции.
- 28) Понятие класса и экземпляра класса (объекта). Поля класса.
- 29) Объявление класса. Спецификаторы доступа к полям и методам.
- 30) Определение методов класса внутри и снаружи объявления класса.

### 6.1. Методические рекомендации при подготовке к зачету

Подготовка студентов к зачету включает проработку лекций, в течение семестра и непосредственную подготовку в дни, предшествующие зачету, включая, конечно, подготовку к коллоквиумам, тестированию, выполнению лабораторных работ и их защиту.

Для подготовки к ответам вопросы зачета (они выдаются в конце семестра) студент должен использовать не только курс лекций, но и основную и дополнительную литературу для выработки умения давать развернутые ответы на поставленные вопросы.

В ходе подготовки к зачету студенту необходимо обращать внимание не только на уровень запоминания, но и на степень понимания изучаемых вопросов. А это достигается не простым заучиванием, а усвоением прочных систематизированных знаний аналитическим мышлением. Следовательно, непосредственная подготовка к зачету должна в разумных пропорциях сочетать и запоминание, и понимание программного материала.

### 6.2. Список основных вопросов к устному экзамену

#### 6 семестр

- 1) Понятие класса и экземпляра класса (объекта). Объявление класса. Спецификаторы доступа к полям и методам. Поля класса.
- 2) Определение методов класса внутри и снаружи объявления класса. Отличие методов класса от обычных функций, указатель this. Константные методы.
- 3) Конструкторы. Конструктор по умолчанию. Список инициализации конструктора. Конструктор копии и оператор присваивания. Защита объектов от копирования.
- 4) Статические поля. Статические константы в классе. Статические методы.
- 5) Владение ресурсами, конструкторы и деструктор, идиома RAII.
- 6) Перегрузка операций. Перегрузка операций operator[] и operator(). Операторы приведения типов.
- 7) Наследование классов. Терминология, применяемая для описания иерархий наследования. Доступ к полям и методам базового класса при наследовании. Спецификаторы доступа при наследовании.
- 8) Переопределение методов базового класса. Реализация конструкторов и деструкторов при наследовании. Статические элементы при наследовании. Наследование и вложенные классы.

- 9) Проблема динамической идентификации типов. Виртуальные функции. Статическое и динамическое связывание. Виртуальный деструктор. Чистые виртуальные функции и абстрактные классы.
- 10) Множественное наследование. Виртуальные базовые классы.
- 11) Механизм RTTI. Применение операции `dynamic_cast`. Операция `typeid`.
- 12) Шаблоны функций и классов. Инстанцирование и работа с шаблонными объектами. Параметры шаблонов. Эквивалентность типов.
- 13) Шаблоны функций. Инстанцирование шаблонных функций. Аргументы шаблонных функций.
- 14) Параметры шаблонов по умолчанию. Полная и частичная специализации шаблонов. Наследование шаблонов.
- 15) Обработка исключений при помощи кодов ошибок и недостатки данного подхода.
- 16) Механизм исключений. Генерация исключений. Перехват исключений, блок `try-catch`. Повторная генерация исключений.
- 17) Механизм исключений. Передача исключений из вложенного блока. Необработанные исключения. Спецификация исключений.
- 18) Организация стандартной библиотеки языка C++. Взаимодействие контейнеров, алгоритмов и итераторов.
- 19) Стандартная библиотека языка C++. Классификация контейнеров. Отличия последовательных и ассоциативных контейнеров.
- 20) Стандартная библиотека языка C++. Функциональные объекты. Сравнение функциональных объектов и обычных функций.
- 21) Стандартная библиотека языка C++. Контейнеры стандартной библиотеки. Требования к элементам контейнера. Общие операции контейнеров. Сравнение контейнеров.
- 22) Динамический массив (`vector`). Устройство контейнера. Сложность операций. Размер и емкость. Сравнение с массивами в стиле языка C.
- 23) Двухнаправленная очередь. Устройство контейнера. Сложность операций. Управление памятью и целостность итераторов.
- 24) Список. Устройство контейнера. Сравнение с контейнером `vector`. Специфические операции со списками.
- 25) Множество и мультимножество. Внутреннее устройство. Требования к критерию сортировки. Вычислительная сложность операций с множествами, функции поиска.
- 26) Отображение и мультиотображение. Внутреннее устройство. Сравнение с множествами. Сложность операций. Отображения в качестве ассоциативных массивов.
- 27) Адаптеры последовательных контейнеров. Стек. Основной интерфейс стека.
- 28) Адаптеры последовательных контейнеров. Очередь. Основной интерфейс очереди.
- 29) Адаптеры последовательных контейнеров. Очередь с приоритетом. Интерфейс очереди с приоритетом.
- 30) Битовые поля. Операции с битовыми полями. Методы класса `bitset`.
- 31) Итераторы стандартной библиотеки. Категории итераторов. Итераторы ввода. Итераторы вывода.
- 32) Итераторы стандартной библиотеки. Категории итераторов. Прямые, двухнаправленные итераторы. Итераторы произвольного доступа.
- 33) Итераторы стандартной библиотеки. Категории итераторов. Функции `advance()`, `distance()`, `iter_swap()`.
- 34) Итераторы стандартной библиотеки. Обратные итераторы.
- 35) Итераторы вставки. Разновидности итераторов вставки.
- 36) Поточные итераторы. Итераторы входного и выходного потоков.
- 37) Строки STL (класс `string`). Обращение к элементам строки. Размер и емкость строки.
- 38) Строки STL (класс `string`). Операции присваивания, вставки, замены и удаления.
- 39) Строки STL (класс `string`). Подстроки и конкатенация. Поиск символов и строк. Строки в качестве контейнеров. Строки и итераторы.
- 40) Ввод/вывод с помощью потоковых объектов.

## 6.2. Методические рекомендации при подготовке к экзамену

Подготовка студентов к экзамену включает проработку лекций, в течении семестра и

непосредственную подготовку в дни, предшествующие экзамену, включая, конечно, подготовку к коллоквиумам, тестированию, выполнению лабораторных работ и их защиту.

Для подготовки к ответам на экзаменационные вопросы (они выдаются в конце семестра) студент должен использовать не только курс лекций, но и основную и дополнительную литературу для выработки умения давать развернутые ответы на поставленные вопросы.

В ходе подготовки к экзамену студенту необходимо обращать внимание не только на уровень запоминания, но и на степень понимания изучаемых вопросов. А это достигается не простым заучиванием, а усвоением прочных систематизированных знаний аналитическим мышлением. Следовательно, непосредственная подготовка к экзамену должна в разумных пропорциях сочетать и запоминание, и понимание программного материала.

### 6.3. Распределение баллов текущего, рубежного контроля и экзамена

| №         |                                        | Общая сумма          | 1-я точка                            | 2-я точка           | 3 точка             |
|-----------|----------------------------------------|----------------------|--------------------------------------|---------------------|---------------------|
| <b>1.</b> | <b>Текущий контроль</b>                |                      |                                      |                     |                     |
|           | посещение занятий                      | 10<br>баллов         | 3<br>балла                           | 3<br>балла          | 4<br>балла          |
|           | выполнение и защита лабораторных работ | 21<br>балл           | 7<br>баллов                          | 7<br>баллов         | 7<br>баллов         |
| <b>2.</b> | <b>Рубежный контроль</b>               |                      |                                      |                     |                     |
|           | тестирование                           | 15<br>баллов         | 5<br>баллов                          | 5<br>баллов         | 5<br>баллов         |
|           | коллоквиум                             | 24<br>балла          | 8<br>баллов                          | 8<br>баллов         | 8<br>баллов         |
|           | <b>Итого</b>                           | <b>70<br/>баллов</b> | <b>23<br/>балла</b>                  | <b>23<br/>балла</b> | <b>24<br/>балла</b> |
| <b>3.</b> | <b>Экзамен</b>                         | <b>30<br/>баллов</b> | <b>min – 15, max – 30<br/>баллов</b> |                     |                     |

### 6.4. Критерии оценивания

При освоении дисциплины формируются компетенции ОПК-4. Указанные компетенции формируются в соответствии со следующими этапами:

- формирование и развитие теоретических знаний, предусмотренных указанными компетенциями (лекционные занятия, самостоятельная работа студентов);
- приобретение и развитие практических умений, предусмотренных компетенциями (лабораторные работы, самостоятельная работа студентов);
- закрепление теоретических знаний, умений и практических навыков, предусмотренных компетенциями (лабораторные работы, практики, выпускная квалификационная работа).

Сформированность компетенций в рамках освоения данной дисциплины оценивается по трехуровневой шкале:

- базовый уровень (**оценка «удовлетворительно»**) является обязательным для всех обучающихся по завершении освоения дисциплины;
- продвинутый уровень (**оценка «хорошо»**) характеризуется превышением минимальных характеристик сформированности компетенций по завершении освоения дисциплины;
- высокий уровень (**оценка «отлично»**) характеризуется максимально возможной выраженностью компетенций и является важным качественным ориентиром для самосовершенствования.

**Критерии оценки качества освоения дисциплины приведены в приложении 2.**

## 7. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и опыта деятельности

Таблица 6. Результаты освоения учебной дисциплины, подлежащие проверке.

| Результаты обучения (компетенции)                                                                                                                                                                                                                                                                                                                                                                                                 | Основные показатели оценки результатов обучения                                                                                                                                                                                                                                                                                                                                                               | Вид оценочного материала                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p><b>ОПК-4.</b> Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения</p> <p><b>Код и наименование индикатора достижения компетенции</b></p> <p><b>ОПК-4.1.</b> Способен разрабатывать алгоритмы решения задач своей профессиональной деятельности.</p> <p><b>ОПК-4.2.</b> Способен разрабатывать программное обеспечение для решения задач своей профессиональной деятельности.</p> | <p><b>Знать</b> основы программирования: языки программирования (Python, Java, C++, и др.), принципы работы компьютера на низком уровне (процессор, память, устройства ввода/вывода).</p> <p><b>Уметь</b> работать с разными инструментами и средами разработки (IDE, системы контроля версий, библиотеки и т.д.)</p> <p><b>Владеть</b> навыками работы с базами данных и написание эффективных запросов.</p> | <p>Выполнение и защита лабораторных работ; типовые оценочные материалы для устного опроса (<i>раздел 5.1.1</i>); типовые тестовые задания (<i>раздел 5.2.</i>); типовые оценочные материалы к экзамену (<i>раздел 6.</i>).</p> <p>Выполнение и защита лабораторных работ; типовые оценочные материалы для устного опроса (<i>раздел 5.1.1</i>); типовые тестовые задания (<i>раздел 5.2.</i>); типовые оценочные материалы к экзамену (<i>раздел 6.</i>).</p> <p>Выполнение и защита лабораторных работ; типовые оценочные материалы для устного опроса (<i>раздел 5.1.1</i>); типовые тестовые задания (<i>раздел 5.2.</i>); типовые оценочные материалы к экзамену (<i>раздел 6.</i>).</p> |

## 8. Учебно-методическое обеспечение дисциплины (модуля)

### Основная литература

1. Мейер, Б. Основы объектно-ориентированного проектирования : учебник / Б. Мейер. — 2-е изд. — Москва : ИНТУИТ, 2016. — 765 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/100305>
2. Мейер, Б. Объектно-ориентированное программирование и программная инженерия : учебное пособие / Б. Мейер. — 2-е изд. — Москва : ИНТУИТ, 2016. — 285 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/100271>
3. Логинова Ф.С. Объектно-ориентированные методы программирования. Институт электронного обучения Санкт-Петербургского университета технологий управления и экономики. 2012. 208 с. <https://reader.lanbook.com/book/64040#2>

### Дополнительная литература

1. Городняя Л. В. Парадигма программирования: учебное пособие для вузов. Издательство "Лань". 2021. 232 с.
2. Коломийцева, С. В. Объектно-ориентированный анализ и проектирование : учебное пособие / С. В. Коломийцева. — Хабаровск : ДВГУПС, 2022. — 82 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/339566>
3. Унгер, А. Ю. Шаблоны объектно-ориентированного проектирования в языке C++ : учебное пособие / А. Ю. Унгер. — Москва : РТУ МИРЭА, 2022. — 67 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/310838>
4. Достовалов, Д. Н. Объектно-ориентированный анализ и проектирование. Задачи и примеры на C++ : учебное пособие / Д. Н. Достовалов, О. В. Лауферман. — Новосибирск : НГТУ, 2022. — 74 с. — ISBN 978-5-7782-4708-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/306206>

## Периодические издания

Перечень периодических изданий, получаемых библиотекой КБГУ, в которых студент может ознакомиться с современными достижениями в области электроники, микро и нано-электроники:

- Физика. (Физика полупроводниковых проводников и диэлектриков, квантовая электроника). Известия ВУЗов.
- Электроника.
- Физика и техника полупроводников.
- Микроэлектроника.
- Квантовая электроника.
- Радиоэлектроника
- Материалы электронной техники.
- Физика твердого тела
- Известия вузов.

## Интернет-ресурсы

1. <http://lib.kbsu.ru/> - Библиотека КБГУ.
2. <http://www.garant.ru/> - Справочная правовая система «Гарант».
3. <http://www.consultant.ru/> -Справочная правовая система «КонсультантПлюс».
4. <http://www.studmedlib.ru/> - ЭБС «Консультант студента»
5. [http://www.ph4s.ru/book\\_electronika.html](http://www.ph4s.ru/book_electronika.html) - Образовательный проект А.Н. Варгина
6. <http://www.Russianelectronics.ru> -портал «Время электроники»;
7. <http://www.platan.ru> – каталог электронных компонентов;
8. <https://www.sciencedirect.com/> - Полнотекстовая база данных ScienceDirect.

**Перечень актуальных электронных информационных баз данных, к которым обеспечен доступ пользователям КБГУ (2024-2025 уч.г.)**

| №п/п                           | Наименование электронного ресурса | Краткая характеристика                                                                                                                                                                           | Адрес сайта                                                 | Наименование организации-владельца; реквизиты договора                                                      | Условия доступа                                |
|--------------------------------|-----------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------|------------------------------------------------|
| <b>РЕСУРСЫ ДЛЯ ОБРАЗОВАНИЯ</b> |                                   |                                                                                                                                                                                                  |                                                             |                                                                                                             |                                                |
| 1.                             | ЭБС «Лань»                        | Электронные версии книг ведущих издательств учебной и научной литературы (в том числе университетских издательств), так и электронные версии периодических изданий по различным областям знаний. | <a href="https://e.lanbook.com/">https://e.lanbook.com/</a> | ООО «ЭБС ЛАНЬ»<br>(г. Санкт-Петербург)<br>Договор №55/ЕП-223<br>от 08.02.2024 г.<br>Активен до 15.02.2025г. | Полный доступ (регистрация по IP-адресам КБГУ) |

|                          |                                                      |                                                                                                                                                                                                                                                        |                                                             |                                                                                                                                        |                                                       |
|--------------------------|------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------|
| 2.                       | <b>Национальная электронная библиотека РГБ</b>       | Объединенный электронный каталог фондов российских библиотек, содержащий 4 331 542 электронных документов образовательного и научного характера по различным отраслям знаний                                                                           | <a href="https://rusneb.ru/">https://rusneb.ru/</a>         | ФГБУ «Российская государственная библиотека»<br>Договор №101/НЭБ/1666-п от 10.09.2020г.<br>Бессрочный                                  | Авторизованный доступ с АРМ библиотеки (ИЦ, ауд.№115) |
| 3.                       | <b>ЭБС «IPSMART»</b>                                 | 107831 публикаций, в т.ч.: 19071 – учебных изданий, 6746 – научных изданий, 700 коллекций, 343 журнала ВАК, 2085 аудиоизданий.                                                                                                                         | <a href="http://iprbookshop.ru/">http://iprbookshop.ru/</a> | ООО «Ай Пи Эр Медиа» (г. Красногорск, Московская обл.)<br><b>№156/24П</b><br>от 04.04.2024 г.<br>срок предоставления лицензии: 12 мес. | Полный доступ (регистрация по IP-адресам КБГУ)        |
| 4.                       | <b>ЭБС «Юрайт» для ВО</b>                            | Электронные версии 8000 наименований учебной и научной литературы издательств «Юрайт» для ВО и электронные версии периодических изданий по различным областям знаний.                                                                                  | <a href="https://urait.ru/">https://urait.ru/</a>           | ООО «Электронное издательство ЮРАЙТ» (г. Москва)<br><b>Договор №54/ЕП-223</b><br>От 08.02.2024 г.<br>Активен по 28.02.2025 г.          | Полный доступ (регистрация по IP-адресам КБГУ)        |
| <b>РЕСУРСЫ ДЛЯ НАУКИ</b> |                                                      |                                                                                                                                                                                                                                                        |                                                             |                                                                                                                                        |                                                       |
| 5.                       | <b>Научная электронная библиотека (НЭБ РФФИ)</b>     | Электр. библиотека научных публикаций - около 4000 иностранных и 3900 отечественных научных журналов, рефераты публикаций 20 тыс. журналов, а также описания 1,5 млн. зарубежных и российских диссертаций; 2800 росс. журналов на безвозмездной основе | <a href="http://elibrary.ru">http://elibrary.ru</a>         | ООО «НЭБ»<br>Лицензионное соглашение №14830 от 01.08.2014г.<br>Бессрочное                                                              | Полный доступ                                         |
| 6.                       | <b>Президентская библиотека им. Б.Н. Ельцина</b>     | Более 500 000 электронных документов по истории Отечества, российской государственности, русскому языку и праву                                                                                                                                        | <a href="http://www.prlib.ru">http://www.prlib.ru</a>       | ФГБУ «Президентская библиотека им. Б.Н. Ельцина» (г. Санкт-Петербург)<br><b>Соглашение от 15.11.2016г.</b><br>Бессрочный               | Авторизованный доступ из библиотеки (ауд. №115, 214)  |
| 7.                       | <b>Polpred.com. Новости. Обзор СМИ. Россия и за-</b> | Обзор СМИ России и зарубежья. Полные тексты + аналитика из 600                                                                                                                                                                                         | <a href="http://polpred.com">http://polpred.com</a>         | ООО «Полпред справочники»<br>Безвозмездно (без официального до-                                                                        | Доступ по IP-адресам КБГУ                             |

|  |         |                        |  |         |  |
|--|---------|------------------------|--|---------|--|
|  | рубежье | изданий по 53 отраслям |  | говора) |  |
|--|---------|------------------------|--|---------|--|

## 9. Программное обеспечение современных информационно-коммуникационных технологий

1. Студенты имеют доступ через Интернет доступ к единому образовательному portalу, где в открытом доступе имеются ресурсы учебно-методической литературы, являющиеся разработками ведущих вузов России.
2. Для рейтингового контроля используется система компьютерного тестирования на базе программного обеспечения Moodle.
3. При выполнении лабораторного практикума студенты проводят обработку экспериментальных данных с применением российских программных сред.
4. В рамках обеспечения применения компьютерных технологий в образовательном процессе имеются специализированные компьютерные классы с современным программным обеспечением и имеющим выход в Интернет.

## 10. Материально-техническое обеспечение дисциплины

Для реализации рабочей программы дисциплины имеются учебные аудитории для проведения учебных занятий, оснащенные оборудованием и техническими средствами обучения, а также помещения для самостоятельной работы, оснащенные компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду КБГУ.

Перечень материально-технического обеспечения дисциплины включает в себя:

- **учебная аудитория для проведения учебных занятий №238**, оснащена оборудованием и техническими средствами обучения (ноутбук, проектор, интерактивная доска, доска стационарная). Комплект учебной мебели – 24 посадочных мест.

- **компьютерный класс для проведения лабораторных и практических занятий, текущего контроля, промежуточной аттестации – 324**, оснащен комплектом учебной мебели, компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду КБГУ. 14 посадочных мест.

- Компьютерная техника обеспечена необходимым комплектом лицензионного и свободно распространяемого программного обеспечения, в том числе отечественного производства. Электронно-библиотечные системы и электронная информационно-образовательная среда КБГУ обеспечивают доступ (удаленный доступ) обучающимся, к современным профессиональным базам данных и информационным справочным системам.

- **помещение для самостоятельной работы - 311, Электронный читальный зал №3. Читальный зал естественных и технических наук**, оснащен комплектом учебной мебели, компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду КБГУ. 22 посадочных места. Компьютерная техника обеспечена необходимым комплектом лицензионного и свободно распространяемого программного обеспечения, в том числе отечественного производства. Электронно-библиотечные системы и электронная информационно-образовательная среда КБГУ обеспечивают доступ (удаленный доступ) обучающимся, к современным профессиональным базам данных и информационным справочным системам.

- **помещение для самостоятельной работы – 115 Электронный читальный зал №1**, оснащен комплектом учебной мебели, компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду КБГУ. 28 посадочных мест. Компьютерная техника обеспечена необходимым комплектом лицензионного и свободно распространяемого программного обеспечения, в том числе отечественного производства. Электронно-библиотечные системы и электронная информационно-образовательная среда КБГУ обеспечивают доступ (удаленный доступ) обучающимся, к современным профессиональным базам данных и информационным справочным системам.

Мультимедийная презентация, сопровождающая лекцию, позволяет преподавателю акцентировать внимание студенческой аудитории на ключевых вопросах лекции.

Дисциплина обеспечена:



- тестовым материалами в электронной обучающей системе «Moodle» (Открытый университет);
- книжным фондом библиотеки;
- электронными версиями лекций и учебников.

Студенты имеют доступ через Интернет доступ к электронной обучающей системе «Moodle» (Открытый университет), которая позволяет размещать электронные учебные курсы в свободном доступе для студентов университета.

Для проведения занятий имеется необходимый комплект лицензионного и свободно распространяемого программного обеспечения, в том числе отечественного производства.

#### **Список лицензионного программного обеспечения**

1. Антивирусное средство для защиты ПК (продление) Kaspersky Endpoint Security.
2. Система оптического распознавания текста (продление) SETERE OCR
3. Многофункциональный редактор (продление) Content Reader PDF 15 Business.
4. РЕД ОС. Техническая поддержка для образовательных учреждений на 1 год. Конфигурация: Сервер. Стандартная редакция. Базовый уровень.
5. РЕД ОС. Техническая поддержка для образовательных учреждений на 1 год. Конфигурация: Рабочая станция. Стандартная редакция. Базовый уровень.
6. Российский кроссплатформенный пакет приложений для совместной работы с офисными документами Р7-Офис.
7. Многофункциональный кроссплатформенный графический редактор AliveColors Business.
8. Комплекс программ автоматизации решения задач конструкторско-технологической подготовки производства и бизнес-процессов САПР Грация.
9. Предоставление неисключительных прав на использование программного обеспечения Системы Spider Project Professional.
10. Программный продукт, основанный на исходном коде свободного проекта Wine, предназначенный для запуска Windows-приложений на операционных системах семейства Linux.

#### свободно распространяемые программы:

7Zip;

DjVu Plug-in;

Система локальной сети КБГУ предоставляет возможность одновременной работы большого количества пользователей как в локальной сети вуза, так и через сеть «Интернет» с соблюдением требований информационной безопасности и ограничением доступа к информации. Электронная информационно – образовательная среда КБГУ позволяет осуществлять работу обучающихся из любой точки доступа, в том числе извне вуза.

#### **Особенности реализации дисциплины для инвалидов и лиц с ограниченными возможностями здоровья**

Для студентов с ограниченными возможностями здоровья созданы специальные условия для получения образования. В целях доступности получения высшего образования по образовательным программам инвалидами и лицами с ограниченными возможностями здоровья университетом обеспечивается:

1. Альтернативная версия официального сайта в сети «Интернет» для слабовидящих;
2. Для инвалидов с нарушениями зрения (слабовидящие, слепые) - присутствие ассистента, оказывающего обучающемуся необходимую помощь, дублирование вслух справочной информации о расписании учебных занятий; наличие средств для усиления остаточного зрения, брайлевской компьютерной техники, видеоувеличителей, программ не визуального доступа к информации, программ-синтезаторов речи и других технических средств приема-передачи учебной информации в доступных формах для студентов с нарушениями зрения;
3. Для инвалидов и лиц с ограниченными возможностями здоровья по слуху (слабослышащие, глухие) – звукоусиливающая аппаратура, мультимедийные средства и другие технические средства приема-передачи учебной информации в доступных формах;
4. Для инвалидов и лиц с ограниченными возможностями здоровья, имеющих нарушения опорно-двигательного аппарата, созданы материально-технические условия обеспечивающие возможность беспрепятственного доступа обучающихся в учебные помещения, объекты питания, туалетные и другие помещения университета, а также пребывания в указанных помещениях

(наличие расширенных дверных проемов, поручней и других приспособлений).

Обучающиеся из числа лиц с ограниченными возможностями здоровья обеспечены электронными образовательными ресурсами в формах, адаптированных к ограничениям их здоровья.

Обучающимся с ограниченными возможностями здоровья предоставляются специальные учебники и учебные пособия, иная учебная литература, специальные технические средства обучения коллективного и индивидуального пользования, предоставление услуг ассистента (помощника), оказывающего обучающимся необходимую техническую помощь, а также услуги сурдопереводчиков и тифлосурдопереводчиков.

а) для слабовидящих:

- на экзамене присутствует ассистент, оказывающий студенту необходимую техническую помощь с учетом индивидуальных особенностей (он помогает занять рабочее место, передвигаться, прочитать и оформить задание, в том числе записывая под диктовку);

- задания для выполнения, а также инструкция о порядке проведения зачета/экзамена оформляются увеличенным шрифтом;

- задания для выполнения на экзамене зачитываются ассистентом;

- письменные задания выполняются на бумаге, надиктовываются ассистенту;

- обеспечивается индивидуальное равномерное освещение не менее 300 люкс;

- студенту для выполнения задания при необходимости предоставляется увеличивающее устройство;

в) для глухих и слабослышащих:

- на зачете/экзамене присутствует ассистент, оказывающий студенту необходимую техническую помощь с учетом индивидуальных особенностей (он помогает занять рабочее место, передвигаться, прочитать и оформить задание, в том числе записывая под диктовку);

- зачет/экзамен проводится в письменной форме;

- обеспечивается наличие звукоусиливающей аппаратуры коллективного пользования, при необходимости поступающим предоставляется звукоусиливающая аппаратура индивидуального пользования;

- по желанию студента экзамен может проводиться в письменной форме;

д) для лиц с нарушениями опорно-двигательного аппарата (тяжелыми нарушениями двигательных функций верхних конечностей или отсутствием верхних конечностей):

- письменные задания выполняются на компьютере со специализированным программным обеспечением или надиктовываются ассистенту;

- по желанию студента экзамен проводится в устной форме.

Лист изменений (дополнений) в рабочей программе дисциплины (модуля) «Объектно-ориентированное программирование» по специальности 12.05.01 Электронные и опто-электронные приборы и системы специального назначения, специализация: «Опто-электронные информационно-измерительные приборы и системы» на 2025 – 2026 учебный год

| № п/п | Элемент (пункт) РПД | Перечень вносимых изменений | Примечание |
|-------|---------------------|-----------------------------|------------|
|       |                     |                             |            |
|       |                     |                             |            |
|       |                     |                             |            |
|       |                     |                             |            |
|       |                     |                             |            |
|       |                     |                             |            |

*Обсуждена и рекомендована на заседании кафедры  
электроники и цифровых информационных техно-  
логий, протокол № \_\_\_\_\_  
от «\_\_\_\_\_» «\_\_\_\_\_» 2024 г.*

Заведующий кафедрой

\_\_\_\_\_/ Р.Ш. Тешев /\_\_\_\_\_  
подпись                      расшифровка подписи                      дата

Критерии оценки качества освоения дисциплины

| Код компетенции                                                                                                                          | РЕЗУЛЬТАТ ОБУЧЕНИЯ по дисциплине (модулю)                                                                                                                                   | КРИТЕРИИ И ПОКАЗАТЕЛИ ОЦЕНИВАНИЯ РЕЗУЛЬТАТА ОБУЧЕНИЯ |                                                                                                                                                                                            |                                                                                                                                                                                          |                                                                                                                                                                                                  |                                                                                                                                                                                                          |
|------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                                                                                                          |                                                                                                                                                                             | Шкала по традиционной пятибалльной системе           |                                                                                                                                                                                            |                                                                                                                                                                                          |                                                                                                                                                                                                  |                                                                                                                                                                                                          |
|                                                                                                                                          |                                                                                                                                                                             | недопуск                                             | неудовлетворительно                                                                                                                                                                        | Базовый уровень удовлетворительно /диф. зачет                                                                                                                                            | Продвинутый уровень хорошо/ диф. зачет                                                                                                                                                           | Высокий уровень отлично/ диф. зачет                                                                                                                                                                      |
|                                                                                                                                          |                                                                                                                                                                             | Шкала по балльно-рейтинговой системе                 |                                                                                                                                                                                            |                                                                                                                                                                                          |                                                                                                                                                                                                  |                                                                                                                                                                                                          |
|                                                                                                                                          |                                                                                                                                                                             | 0 – 35                                               | 36 – 60                                                                                                                                                                                    | 61 – 80                                                                                                                                                                                  | 81 – 90                                                                                                                                                                                          | 91 – 100                                                                                                                                                                                                 |
| ОПК-4. Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения.<br>Код наименования индикатора | Знать основы программирования: языки программирования (Python, Java, C++, и др.), принципы работы компьютера на низком уровне (процессор, память, устройства ввода/вывода). | Не знает                                             | отсутствие знаний о основах программирования: языки программирования (Python, Java, C++, и др.), принципы работы компьютера на низком уровне (процессор, память, устройства ввода/вывода). | неполные знания о основах программирования: языки программирования (Python, Java, C++, и др.), принципы работы компьютера на низком уровне (процессор, память, устройства ввода/вывода). | в целом успешные знания о основах программирования: языки программирования (Python, Java, C++, и др.), принципы работы компьютера на низком уровне (процессор, память, устройства ввода/вывода). | полностью сформированные знания о основах программирования: языки программирования (Python, Java, C++, и др.), принципы работы компьютера на низком уровне (процессор, память, устройства ввода/вывода). |

|                                                                                                                                                                                                                                                                        |                                                                                                                              |            |                                                                                                                                          |                                                                                                                               |                                                                                                                                  |                                                                                                                                          |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------|------------|------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------|
| <p><i>достижения компетенции:</i><br/>ОПК-4.1.<br/>Способен разрабатывать алгоритмы решения задачи своей профессиональной деятельности.</p> <p>ОПК-4.2.<br/>Способен разрабатывать программное обеспечение для решения задачи своей профессиональной деятельности.</p> | <p><b>Уметь</b> работать с разными инструментами и средами разработки (IDE, системы контроля версий, библиотеки и т.д.).</p> | Не умеет   | Отсутствие или частичное умение работать с разными инструментами и средами разработки (IDE, системы контроля версий, библиотеки и т.д.). | недостаточное умение работать с разными инструментами и средами разработки (IDE, системы контроля версий, библиотеки и т.д.). | в целом успешное умение работать с разными инструментами и средами разработки (IDE, системы контроля версий, библиотеки и т.д.). | Полностью сформированное умение работать с разными инструментами и средами разработки (IDE, системы контроля версий, библиотеки и т.д.). |
|                                                                                                                                                                                                                                                                        | <p><b>Владеть</b> навыками работы с базами данных и написание эффективных запросов.</p>                                      | Не владеет | Отсутствие или частичное владение навыками работы с базами данных и написание эффективных запросов.                                      | недостаточное владение навыками работы с базами данных и написание эффективных запросов.                                      | в целом успешное владение навыками работы с базами данных и написание эффективных запросов.                                      | Полностью сформированное владение навыками работы с базами данных и написание эффективных запросов.                                      |