

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«Воронежский государственный технический университет»

УТВЕРЖДАЮ

Декан факультета информационных
технологий и компьютерной безопасности

Бредихин А.В.

23.06.2026



**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ
«Объектно-ориентированное программирование»**

Направление подготовки 09.03.02 Информационные системы и технологии

Профиль Технологии искусственного интеллекта

Квалификация выпускника бакалавр

Нормативный период обучения 4 года

Форма обучения очная

Год начала подготовки 2024

Авторы программы:

Б.Н. Тишуков

директор Воронежского филиала
АО «Неофлекс Консалтинг»

А.М. Нагайцев

Заведующий кафедрой
искусственного интеллекта
и цифровых технологий

П.Ю. Гусев

Руководитель ОПОП

Д.В. Иванов

Воронеж 2026

1. ЦЕЛИ И ЗАДАЧИ ДИСЦИПЛИНЫ

1.1. Цели дисциплины

изучение объектно-ориентированной методологии программирования, изучение основных понятий объектно-ориентированного программирования.

1.2. Задачи освоения дисциплины

- к теоретическим задачам относятся: обучение разработке программного обеспечения в объектно-ориентированной парадигме; изучение основ программирования на языке C++;
- прикладные задачи состоят в приобретении навыков проектирования и разработки программного обеспечения с применением объектно-ориентированного подхода к программированию.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

Дисциплина «Объектно-ориентированное программирование» относится к дисциплинам базовой части блока Б1.

3. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Процесс изучения дисциплины «Объектно-ориентированное программирование» направлен на формирование следующих компетенций:

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

ОПК-4 - Способен участвовать в разработке технической документации, связанной с профессиональной деятельностью с использованием стандартов, норм и правил;

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

Компетенция	Результаты обучения, характеризующие сформированность компетенции
ОПК-2	знать методы и технологии программирования, принципы и определения объектно-ориентированного подхода
	уметь описывать основные структуры данных на основе объектно-ориентированного подхода

	владеть навыками работы в различных средах программирования
ОПК-4	знать виды технической документации поддержки программного кода
	уметь описывать техническую документацию
	владеть навыками оформления программного кода
ОПК-6	знать основные принципы конструирования алгоритмов
	уметь разрабатывать и реализовывать алгоритмы на языке высокого уровня
	владеть методами описания основных классов и алгоритмов на их основе

4. ОБЪЕМ ДИСЦИПЛИНЫ

Общая трудоемкость дисциплины «Объектно-ориентированное программирование» составляет 6 з.е.

Распределение трудоемкости дисциплины по видам занятий

Виды учебной работы	Всего часов	Семестры
		2
Аудиторные занятия (всего)	90	90
В том числе:		
Лекции	36	36
Лабораторные работы (ЛР)	54	54
Самостоятельная работа	90	90
Курсовой проект	+	+
Часы на контроль	36	36
Виды промежуточной аттестации - экзамен	+	+

Общая трудоемкость: академические часы зач.ед.	216 6	216 6
--	----------	----------

5. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

5.1 Содержание разделов дисциплины и распределение трудоемкости по видам занятий

№ п/п	Наименование темы	Содержание раздела	Лекц	Лаб. зан.	СРС	Всего , час
1	Общие сведения, Основы программирования на C++	Введение. Необходимость объектно-ориентированное программирование, характеристики объектно-ориентированных языков Основы программирования на C++. История C++, сравнение с языком C. Структура программы на языке C++. Директивы, комментарии. Переменные. Преобразование типов. Арифметические операции. Обзор интегрированных сред разработки (IDE). Обзор интегрированных IDE и облачных средств разработки.	8	12	18	38
2	Циклы и ветвления, структуры и функции	Циклы и ветвления. Операции отношения. Циклы и ветвления. Логические операции. Приоритеты операций. Структуры. Использование структур, перечислений. Функции в C++	8	12	18	38
3	Объектно-ориентированный подход к разработке программных средств	Введение в классы и объекты. Описание и свойства объектов. Конструкторы, деструкторы классов, объявления статических объектов, переменных и методов в классе. Спецификаторы доступа. Наследование в ООП. Наследование, виды наследования. Абстрактные классы. Множественное наследование. Инкапсуляция и полиморфизм. Принципы SOLID	8	10	18	36

4	Потоки и файлы. Многофайловые программы.	Потоки и файлы. Потокотые классы. Потокотый ввод/вывод дискотых файлов. Указатели файлов. Файловый ввод/вывод с помощью методов. Синхронизация потоков. Многофайловые программы. Создание многофайловой программы. Межфайловое взаимодействие.	6	10	18	34
5	Шаблоны и исключения. Стандартная библиотека шаблонов.	Шаблоны и исключения. Шаблоны функций и классов. Исключения. Стандартная библиотека шаблонов. Последовательные контейнеры. Итераторы. Специальные итераторы. Ассоциативные контейнеры. Хранение пользовательских объектов. Функциональные объекты.	6	10	18	34
Итого			36	54	90	180

5.2 Перечень лабораторных работ

Лабораторная работа № 1. Базовые конструкции языка C++. Управляющие конструкции.

Лабораторная работа № 2. Работа с массивами.

Лабораторная работа № 3. Подпрограммы.

Лабораторная работа № 4. Работа со строками и файлами.

Лабораторная работа № 5. Классы, свойства, индексаторы.

Лабораторная работа № 6. Наследование. Реализация полиморфизма.

Лабораторная работа № 7. Виртуальные функции и полиморфизм.

Лабораторная работа № 8. Обработка исключений.

Лабораторная работа № 9. Регулярные выражения. Шаблоны.

Лабораторная работа № 10. Многопоточное программирование.

Лабораторная работа № 11. Абстракция и инкапсуляция. Реализация принципов SOLID.

6. ПРИМЕРНАЯ ТЕМАТИКА КУРСОВЫХ ПРОЕКТОВ (РАБОТ) И КОНТРОЛЬНЫХ РАБОТ

В соответствии с учебным планом освоение дисциплины предусматривает выполнение курсового проекта во 2 семестре.

Примерная тематика курсового проекта: «Разработка программного обеспечения с использованием объектно-ориентированного подхода (по вариантам)».

Варианты заданий:

- 1 Библиотека
- 2 Складское хозяйство
- 3 Почтовая служба
- 4 Служба такси
- 5 Гостиница
- 6 Заказ товаров
- 7 Туристическое агентство
- 8 Портфолио студента
- 9 Учет успеваемости студентов
- 10 Банк
- 11 Аптека
- 12 Медицинские учреждения
- 13 Магазин
- 14 Салон красоты
- 15 Автосервис
- 16 Пекарня
- 17 Ресторан/Кафе
- 18 Аэропорт

В рамках выполнения курсового проекта студенту необходимо для предметной области, соответствующей выбранному варианту, провести ее анализ, спроектировать информационную систему, основываясь на основных принципах и методах ООП, разработать ее математическую модель и реализовать систему в виде программного продукта на языке C++.

7. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ

7.1. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания

7.1.1 Этап текущего контроля

Результаты текущего контроля знаний и межсессионной аттестации оцениваются по следующей системе:

«аттестован»;

«не аттестован».

Компетенция	Результаты обучения, характеризующие сформированность компетенции	Критерии оценивания	Аттестован	Не аттестован
ОПК-2	знать методы и технологии программирования, принципы и определения объектно-ориентированного подхода	Тестирование Проверка этапов выполнения курсового проекта Результаты проведения коллоквиума Ответы на теоретические вопросы при защите лабораторных работ	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	уметь описывать основные структуры данных на основе объектно-ориентированного подхода	Эффективность использования изученного теоретического материала при выполнении лабораторных работ	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	владеть навыками работы в различных средах программирования	Владение языком программирования C++ при выполнении лабораторных работ	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
ОПК-4	знать виды технической документации поддержки программного кода	Тестирование Проверка этапов выполнения курсового проекта Результаты проведения коллоквиума Ответы на теоретические вопросы при защите лабораторных работ	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	уметь описывать техническую документацию	Эффективность использования изученного теоретического материала при выполнении лабораторных работ	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах

	владеть навыками оформления программного кода	Владение способами описания методики использования программного средства при выполнении лабораторных работ	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
ОПК-6	знать основные принципы конструирования алгоритмов	Тестирование Проверка этапов выполнения курсового проекта Результаты проведения коллоквиума Ответы на теоретические вопросы при защите лабораторных работ	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	уметь разрабатывать и реализовывать алгоритмы на языке высокого уровня	Эффективность использования изученного теоретического материала при выполнении лабораторных работ	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	владеть методами описания основных классов и алгоритмов на их основе	Владение способами описания методики использования программного средства при выполнении лабораторных работ	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах

7.1.2 Этап промежуточного контроля знаний

Результаты промежуточного контроля знаний оцениваются во 2 семестре для очной формы обучения, 2 семестре для заочной формы обучения по четырехбалльной системе:

«отлично»;
«хорошо»;
«удовлетворительно»;
«неудовлетворительно».

Компетенция	Результаты обучения, характеризующие сформированность компетенции	Критерии оценивания	Отлично	Хорошо	Удовл.	Неудовл.

ОПК-2	знать методы и технологии программирования, принципы и определения объектно-ориентированного подхода	Тест	Выполнение теста на 90- 100%	Выполнение теста на 80- 90%	Выполнение теста на 70- 80%	В тесте менее 70% правильных ответов
	уметь описывать основные структуры данных на основе объектно-ориентированного подхода	Тест	Задачи решены в полном объеме и получены верные ответы	Продemonстрирован верный ход решения всех, но не получен верный ответ во всех задачах	Продemonстрирован верный ход решения в большинстве задач	Задачи не решены
	владеть навыками работы в различных средах программирования	Решение прикладных задач в конкретной предметной области	Задачи решены в полном объеме и получены верные ответы	Продemonстрирован верный ход решения всех, но не получен верный ответ во всех задачах	Продemonстрирован верный ход решения в большинстве задач	Задачи не решены
ОПК-4	знать виды технической документации поддержки программного кода	Тест	Выполнение теста на 90- 100%	Выполнение теста на 80- 90%	Выполнение теста на 70- 80%	В тесте менее 70% правильных ответов
	уметь описывать техническую документацию	Решение стандартных практических задач	Задачи решены в полном объеме и получены верные ответы	Продemonстрирован верный ход решения всех, но не получен верный ответ во всех задачах	Продemonстрирован верный ход решения в большинстве задач	Задачи не решены
	владеть навыками оформления программного кода	Решение прикладных задач в конкретной предметной области	Задачи решены в полном объеме и получены верные ответы	Продemonстрирован верный ход решения всех, но не получен верный ответ во всех задачах	Продemonстрирован верный ход решения в большинстве задач	Задачи не решены

ОПК-6	знать основные принципы конструирования алгоритмов	Тест	Выполнение теста на 90- 100%	Выполнение теста на 80-90%	Выполнение теста на 70-80%	В тесте менее 70% правильных ответов
	уметь разрабатывать и реализовывать алгоритмы на языке высокого уровня	Решение стандартных практических задач	Задачи решены в полном объеме и получены верные ответы	Продemonстрирован верный ход решения всех, но не получен верный ответ во всех задачах	Продemonстрирован верный ход решения в большинстве задач	Задачи не решены
	владеть методами описания основных классов и алгоритмов на их основе	Решение прикладных задач в конкретной предметной области	Задачи решены в полном объеме и получены верные ответы	Продemonстрирован верный ход решения всех, но не получен верный ответ во всех задачах	Продemonстрирован верный ход решения в большинстве задач	Задачи не решены

7.2 Примерный перечень оценочных средств (типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности)

7.2.1 Примерный перечень заданий для подготовки к тестированию

- 1 С++ является поддерживает парадигмы программирования:
- А. процедурную
 - Б. объектно-ориентированную
 - В. варианты а А и Б верны**
 - Г. все варианты не верны
- 2 В качестве образца по отношению к объекту выступает:
- А. метод
 - Б. класс**
 - В. операция
 - Г. значение
- 3 Двумя основными компонентами объекта являются _____ и функции, которые _____.
- А. данные; совершают действия над ними**
 - Б. методы; совершают действия над ними
 - В. классы; совершают действия над ними
 - Г. данные; являются объектами
- 4 В С++ функцию, входящую в состав класса, называют:
- А. функция-член класса**
 - Б. оператор класса
 - В. функция класса
 - Г. метод класса
- 5 Защита данных от несанкционированного доступа другими функциями называется _____ .
- А. блокировкой данных
 - Б. полиморфизмом
 - В. инкапсуляцией
 - Г. сокрытием данных**
- 6 Какие из перечисленных ниже причин являются главными для использования объектно-ориентированных языков? (укажите 2 варианта)
- А. возможность создания собственных типов данных**
 - Б. простота операторов объектно-ориентированных языков по сравнению с процедурными языками

В. наличие средств для автокоррекции ошибок в объектно-ориентированных языках

Г. **объектно-ориентированные программы легче концептуализируются**

7 _____ отображают объекты реального мира точнее, чем функции.

А. классы

Б. функции

В. **объекты**

Г. операторы

8 Истинно ли утверждение: программа на C++ в целом схожа с программой на C за исключением незначительных различий в кодировании.

А. да

Б. **нет**

9 Объединение данных и функций называется _____.

А. наследованием

Б. полиморфизмом

В. **инкапсуляцией**

Г. ни один из предложенных вариантов не подходит

10 Если язык обеспечивает возможность создания пользовательских типов данных, то говорят, что язык называется:

А. наследуемым

Б. инкапсулируемым

В. перегруженным

Г. **расширяемым**

7.2.2 Примерный перечень заданий для решения стандартных задач

1 Какое из выражений, выведет значение переменной varbl в поле размером 10 символов?

- А. `cout (setw(10)) << varbl;`
- Б. `cout << setw(10) << varbl;`**
- В. `cout << varbl << setw(10);`
- Г. `cout << varbl (setw(10));`

2 Какой заголовочный файл нужно включить в исходный текст, чтобы использовать объекты cin и cout?

- А. `IOSTREAM`**
- Б. `IOMANIP`
- В. `SETW`
- Г. нет необходимости во включении заголовочного файла

3 Какое из этих выражений получает с клавиатуры числовое значение и присваивает его переменной varbl.

- А. `cin << varbl;`
- Б. `cin >> varbl;`**
- В. `cout >> varbl;`
- Г. `cout << varbl;`

4 Какой заголовочный файл нужно включить в исходный текст, чтобы использовать манипулятор setw?

- А. `IOSTREAM`
- Б. `IOMANIP`**
- В. `SETW`
- Г. нет необходимости во включении заголовочного файла

5 Двумя случаями, когда компилятор обрабатывает разделительные символы, являются _____ и _____ .

- А. директивы препроцессора; заголовочные файлы
- Б. операторы; переменные
- В. операторы; строковые константы
- Г. строковые константы; директивы препроцессора**

6 Верно или неверно следующее утверждение: нет никаких препятствий к использованию переменных разного типа в одном арифметическом выражении?

- А. Верно
- Б. Неверно

7 Значение выражения $11 \% 3$ равно _____ .

- А. 2
- Б. 3
- В. 9
- Г. другое значение

8 Какой оператор, увеличивает значение переменной `varbl` на 23 с одновременным присваиванием?

- А. **`varbl += 23;`**
- Б. `varbl =+ 23;`
- В. `varbl ++= 23;`
- Г. `varbl =++ 23;`

9 На какую величину изменяет значение переменной операция инкремента?

- А. 2
- Б. **1**
- В. -1
- Г. не изменяет

10 Какие значения выведут на экран два указанных оператора, если начальное значение переменной `varbl` равно 20?

```
cout << varbl--;  
cout << ++varbl;
```

- А. 20
- Б. 1921
- В. 2119
- Г. **2020**

7.2.3 Примерный перечень заданий для решения прикладных задач

1 Какой из приведённых циклов `for` будет выводить на экран числа от 100 до 110?

- А. `for (int j = 100; j <= 109; ++j)`
`cout << endl << j;`
- Б. **`for (int j = 100; j <= 110; j++)`**

- cout << endl << j;**
В. for (int j = 100; j <= 111; j —)
 cout << endl << j;
Г. ни один из перечисленных вариантов

2 Блок кода ограничен _____.

- А. **фигурными скобками**
Б. квадратными скобками
В. кавычками
Г. ни один из перечисленных вариантов

3 Переменная, описанная внутри блока, видима:

- А. от точки своего объявления до конца программы
Б. от точки своего объявления до конца функции
В. **от точки своего объявления до конца блока**
Г. внутри функции

4 Какой из приведенных циклов while будет выводить на экран числа от 100 до 110?

- А. int j = 100;
 while(j <= 110);
 cout << endl << j++;
Б. int j = 100;
 cout << endl << j++;
 while(j <= 110);
В. int j = 100;
 while(j = 110)
 cout << endl << j++;
Г. **ни один из перечисленных вариантов**

5 Истинно ли следующее утверждение: операции отношения имеют более высокий приоритет, чем арифметические операции?

- А. истинно
Б. **ложно**

6 Создайте цикл do, который будет выводить на экран числа от 100 до 110.

- А. int j = 100;
 do while(j <= 110);
 cout << endl << j++;
Б. do
 int j = 100;
 cin << endl << j++;
 while(j <= 110);
В. **int j = 100;**

```
do
    cout << endl << j++;
while(j <= 110);
```

Г. ни один из перечисленных вариантов

7 Какое из приведенных ветвление if, выведет сообщение «Больше 10» в случае, если значение переменной varbl больше, чем 10?

А. **if (varbl == 10)**
cout << "Больше 10";

Б. if (varbl > 10)
cout << Больше 10;

В. if (varbl > 10);
cout << "Больше 10";

Г. ни один из перечисленных вариантов

8 Библиотечная функция exit() предназначена для выхода из:

А. цикла, в котором она содержится

Б. блока, в котором она содержится

В. функции, в которой она содержится

Г. **программы, в которой она содержится**

9 Какое из приведенных ветвлений if...else выводит на экран сообщение «Больше 10», если значение переменной varbl больше, чем 10, и сообщение «Меньше 10» в противном случае?

А. if(varbl > 10);
cout << «Больше 10 »;
else

cout << «Меньше 10 »

Б. if(varbl > 10)
cout << «Больше 10 »
else

cout << «Меньше 10 »

В. **if(varbl > 10)**
cout << «Больше 10 »;
else

cout << «Меньше 10 »

Г. ни один из перечисленных вариантов

10 Библиотечная функция getch() (укажите 2 варианта);

А. **возвращает символ в случае нажатия какой-либо из клавиш**

Б. возвращает символ в случае нажатия клавиши Enter

В. **печатает на экране символ, соответствующий нажатой клавише**

Г. не отображает символ на экране

7.2.4 Примерный перечень вопросов для подготовки к зачету

Не предусмотрено учебным планом

7.2.5 Примерный перечень вопросов для подготовки к экзамену

1. Основные понятия ООП.
2. Программирование для Windows.
3. Технология разработки Windows-приложений в Delphi. Визуальное программирование.
4. Основные элементы среды разработки приложений Delphi.
5. Основные визуальные компоненты Delphi. Краткая характеристика.
6. Object Pascal. Вызов функций, передача параметров, возврат результатов, перегрузка функций.
7. Динамическое распределение памяти. Указатели, операции с указателями.
8. Динамические массивы.
9. Самоадресуемые записи, динамические списки.
10. Обработка исключительных ситуаций.
11. Понятие класса, объекта и компонента.
12. Составляющие класса. Объявление класса.
13. Поля и свойства классов.
14. Методы, конструкторы и деструкторы классов.
15. События и обработчики событий классов.
16. Механизм наследования и иерархия классов.
17. Алгоритм построения дерева иерархии классов.
18. Виртуальные методы и полиморфизм.
19. Замещение (перекрытие и перегрузка) методов.
20. Абстрактные методы и классы.
21. Операции с классами, ссылки на классы и информация о классах.
22. Регистрация и обнаружение классов.
23. Создание компонентов во время выполнения, клонирование объектов.
24. Поиск компонентов и уничтожение объектов во время выполнения программы.
25. Двукратное освобождение памяти.
26. Построение собственных компонентов – причины и требования к компонентам.
27. Этапы построения собственных компонентов.
28. Пиктограммы компонентов. Правила именования свойств, полей, методов, модулей, пиктограмм.
29. Типы компонентов Delphi. Фундаментальные классы компонентов библиотеки VCL.
30. Алгоритм построения дерева визуальных объектов формы.
31. **Принципы SOLID**

7.2.6. Методика выставления оценки при проведении промежуточной аттестации

Оценка при проведении промежуточной аттестации учитывает результаты тестирования. Экзамен проводится по экзаменационным билетам, каждый из которых содержит два теоретических вопроса и одно практическое задание, тестирование предполагает получение ответов на 15 вопросов.

1. Оценка «Неудовлетворительно» ставится в случае, если студент дал неправильные ответы на два экзаменационных вопроса, не решил практическое задание и ответил менее чем на 60% тестовых вопросов.

2. Оценка «Удовлетворительно» ставится в случае, если студент неправильно ответил на один экзаменационный вопрос или на все вопросы дал неполные ответы, не решил полностью практическое задание и ответил на 60-80 % тестовых вопросов.

3. Оценка «Хорошо» ставится в случае, если студент правильно ответил на один экзаменационный вопрос или на все вопросы дал неполные ответы, решил полностью практическое задание, продемонстрировал понимание материала, но допустил незначительные ошибки, а также выполнил тест на 80-90%.

4. Оценка «Отлично» ставится, если студент дал обоснованные, глубокие и теоретически правильные ответы на экзаменационные вопросы, решил полностью практическое задание, продемонстрировал понимание материала, а также выполнил тест на 90-100%.

Компетенции считаются сформированными, если в ходе изучения дисциплины выполнены и защищены курсовой проект и лабораторные работы. Выполненные лабораторные работы являются допуском к сдаче экзамена.

7.2.7 Паспорт оценочных материалов

№ п/п	Контролируемые разделы (темы) дисциплины	Код контролируемой компетенции	Наименование оценочного средства
1	Общие сведения, Основы программирования на C++	ОПК-2, ОПК-6	Тест, экзамен, защита лабораторных работ, защита курсового проекта
2	Циклы и ветвления, структуры и функции	ОПК-4, ОПК-6	Тест, экзамен, защита лабораторных работ, защита курсового проекта

3	Объектно-ориентированный подход к разработке программных средств	ОПК-2, ОПК-4	Тест, экзамен, защита лабораторных работ, защита курсового проекта
4	Потоки и файлы. Многофайловые программы	ОПК-2, ОПК-4, ОПК-6	Тест, экзамен, защита лабораторных работ, защита курсового проекта
5	Шаблоны и исключения. Стандартная библиотека шаблонов.	ОПК-2, ОПК-4, ОПК-6	Тест, экзамен, защита лабораторных работ, защита курсового проекта

7.3. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности

Тестирование осуществляется, либо при помощи компьютерной системы тестирования, либо с использованием выданных тест-заданий на бумажном носителе. Время тестирования 45 мин. Затем осуществляется проверка теста экзаменатором и выставляется оценка согласно методики выставления оценки при проведении промежуточной аттестации.

Решение стандартных задач осуществляется с использованием выданных задач на бумажном носителе. Время решения задач 45 мин. Затем осуществляется проверка решения задач экзаменатором и выставляется оценка, согласно методики выставления оценки при проведении промежуточной аттестации.

Решение прикладных задач осуществляется с использованием выданных задач на бумажном носителе. Время решения задач 45 мин. Затем осуществляется проверка решения задач экзаменатором и выставляется оценка, согласно методики выставления оценки при проведении промежуточной аттестации.

Защита курсового проекта осуществляется согласно требованиям, предъявляемым к работе, описанным в методических материалах. Примерное время защиты на одного студента составляет 20 мин.

8. УЧЕБНО МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

8.1. Перечень учебной литературы, необходимой для освоения дисциплины

Холопкина Л.В. Программирование на C++. Учебное пособие / Л.В. Холопкина. – Воронеж, ВГТУ. – 2011. – 184 с.

Холопкина Л.В. Сборник задач по дисциплине “Объектно-ориентированное программирование” / Л.В. Холопкина, Н.И. Гребенникова, М.П. Носачева. – Воронеж: ВГТУ, – 2012.

Лисицин Д.В. Объектно-ориентированное программирование [Электронный ресурс]: конспект лекций/ Д.В. Лисицин— Электрон. текстовые данные.— Новосибирск: Новосибирский государственный технический университет, 2010.— 88 с.— Режим доступа: <http://www.iprbookshop.ru/44970.html>.— ЭБС «IPRbooks».

Новиков П.В. Объектно-ориентированное программирование [Электронный ресурс]: учебно-методическое пособие к лабораторным работам/ П.В. Новиков— Электрон. текстовые данные.— Саратов: Вузовское образование, 2017.— 124 с.— Режим доступа: <http://www.iprbookshop.ru/64650.html>.— ЭБС «IPRbooks».

8.2. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень лицензионного программного обеспечения, ресурсов информационно-телекоммуникационной сети «Интернет», современных профессиональных баз данных и информационных справочных систем:

g++, MS Visual Studio

9. МАТЕРИАЛЬНО-ТЕХНИЧЕСКАЯ БАЗА, НЕОБХОДИМАЯ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА

Для проведения лекционных занятий необходима аудитория, оснащенная проектором

Проведение лабораторных работ проводятся в специализированной лаборатории

10. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ (МОДУЛЯ)

По дисциплине «Объектно-ориентированное программирование» читаются лекции, проводятся лабораторные работы, выполняется курсовой проект.

Основой изучения дисциплины являются лекции, на которых излагаются наиболее существенные и трудные вопросы, а также вопросы, не нашедшие отражения в учебной литературе.

Лабораторные работы выполняются на лабораторном оборудовании в соответствии с методиками, приведенными в указаниях к выполнению работ.

Методика выполнения курсового проекта изложена в учебно-методическом пособии. Выполнять этапы курсового проекта должны своевременно и в установленные сроки.

Контроль усвоения материала дисциплины производится проверкой курсового проекта, защитой курсового проекта.

Вид учебных занятий	Деятельность студента
Лекция	Написание конспекта лекций: кратко, схематично, последовательно фиксировать основные положения, выводы, формулировки, обобщения; помечать важные мысли, выделять ключевые слова, термины. Проверка терминов, понятий с помощью энциклопедий, словарей, справочников с выписыванием толкований в тетрадь. Обозначение вопросов, терминов, материала, которые вызывают трудности, поиск ответов в рекомендуемой литературе. Если самостоятельно не удастся разобраться в материале, необходимо сформулировать вопрос и задать преподавателю на лекции или на практическом занятии.
Лабораторная работа	Лабораторные работы позволяют научиться применять теоретические знания, полученные на лекции при решении конкретных задач. Чтобы наиболее рационально и полно использовать все возможности лабораторных для подготовки к ним необходимо: следует разобрать лекцию по соответствующей теме, ознакомиться с соответствующим разделом учебника, проработать дополнительную литературу и источники, решить задачи и выполнить другие письменные задания.

Самостоятельная работа	Самостоятельная работа студентов способствует глубокому усвоению учебного материала и развитию навыков самообразования. Самостоятельная работа предполагает следующие составляющие: - работа с текстами: учебниками, справочниками, дополнительной литературой, а также проработка конспектов лекций; - выполнение домашних заданий и расчетов; - работа над темами для самостоятельного изучения; - участие в работе студенческих научных конференций, олимпиад; - подготовка к промежуточной аттестации.
Подготовка к промежуточной аттестации	Готовиться к промежуточной аттестации следует систематически, в течение всего семестра. Интенсивная подготовка должна начинаться не позднее, чем за месяц-полтора до промежуточной аттестации. Данные перед экзаменом, экзаменом три дня эффективнее всего использовать для повторения и систематизации материала.

11 Лист регистрации изменений

№ п/п	Перечень вносимых изменений	Дата внесе- ния измене- ний	Подпись заведующего ка- федрой, ответственной за реализацию ОПОП
1	Актуализированы разделы 5, 7 и 8. Актуализация выполнена с привлече- ние специалистов из АО Неофлекс Консалтинг, в соответствии с согла- шением № 2705/2026 от 27 мая 2026 г.	23.06.2026	