

**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

 /П. Ю. Гусев/

24 января 2023 г.



РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

«Среды визуального программирования»

Направление подготовки 09.03.01 Информатика и вычислительная техника

Профиль Технологии интеллектуальных автоматизированных систем

Квалификация выпускника бакалавр

Нормативный период обучения 4 года / 4 года и 11 м.

Форма обучения очная / заочная

Год начала подготовки 2023

Автор программы

А.С. Троценко

Заведующий кафедрой
Компьютерных
интеллектуальных
технологий проектирования

М.И. Чижов

Руководитель ОПОП

М.И. Чижов

Воронеж 2023

1. ЦЕЛИ И ЗАДАЧИ ДИСЦИПЛИНЫ

1.1. Цели дисциплины

Приобретение базовых знаний и навыков программирования, проектирование и разработка приложений с применением объектно-ориентированного подхода, приобретение навыков, позволяющих будущим специалистам вести успешную разработку специализированного программного обеспечения в тех областях и сферах деятельности, в которых они будут трудиться.

Изучение дисциплины должно способствовать формированию у студентов основ научного мышления, в том числе: владение основными методами, способами и инструментами создания программного обеспечения, использования для решения практических задач.

1.2. Задачи освоения дисциплины

- 1) ознакомление с тенденцией развития программного обеспечения;
- 2) обучение разработке алгоритмов на основе объектно-ориентированного подхода в визуальных средах разработки;
- 3) освоение языка программирования C++, а также стандартов кодирования, спецификаций и последующих решений на их основе;
- 4) работа с современными программными инструментальными средствами разработки.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

Дисциплина «Среды визуального программирования» относится к дисциплинам обязательной части блока Б1.

3. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Процесс изучения дисциплины «Среды визуального программирования» направлен на формирование следующих компетенций:

ОПК-8 - Способен разрабатывать алгоритмы и программы, пригодные для практического применения;

ОПК-9 - Способен осваивать методики использования программных средств для решения практических задач.

Компетенция	Результаты обучения, характеризующие сформированность компетенции
ОПК-8	Знать основные объекты представления данных (классы) и методы их обработки, а также способы реализации
	Уметь реализовывать алгоритмы на языке высокого уровня
	Владеть навыками работы в различных средах программирования
ОПК-9	Знать методы и технологии программирования

	Уметь описывать основные структуры данных на основе объектно-ориентированного подхода
	Владеть методами описания основных классов и алгоритмов на их основе

4. ОБЪЕМ ДИСЦИПЛИНЫ

Общая трудоемкость дисциплины «Среды визуального программирования» составляет 5 з.е.

Распределение трудоемкости дисциплины по видам занятий
очная форма обучения

Виды учебной работы	Всего часов	Семестры
		3
Аудиторные занятия (всего)	90	90
В том числе:		
Лекции	36	36
Лабораторные работы (ЛР)	54	54
Самостоятельная работа	54	54
Курсовой проект	+	+
Часы на контроль	36	36
Виды промежуточной аттестации - экзамен	+	+
Общая трудоемкость:		
академические часы	180	180
зач.ед.	5	5

заочная форма обучения

Виды учебной работы	Всего часов	Семестры
		3
Аудиторные занятия (всего)	6	6
В том числе:		
Лекции	2	2
Лабораторные работы (ЛР)	4	4
Самостоятельная работа	165	165
Курсовой проект	+	+
Часы на контроль	9	9
Виды промежуточной аттестации - экзамен	+	+
Общая трудоемкость:		
академические часы	180	180
зач.ед.	5	5

5. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

5.1 Содержание разделов дисциплины и распределение трудоемкости по видам занятий

очная форма обучения

№ п/п	Наименование темы	Содержание раздела	Лекц	Лаб. зан.	СРС	Всего, час
1	Интегрированные среды и средства разработки на языке программирования высокого уровня	Введение. Анализ языков программирования и обзор сред разработки программного обеспечения. Возможности и производительность современных компиляторов, сборщиков проектов. Среды разработки. Интерфейс и конфигурация интегрированной среды разработки (IDE) под решение вычислительных задач. Разработка консольных проектов и приложений с графическим интерфейсом.	4	4	4	12
2	Базовые конструкции и ключевые слова языка программирования высокого уровня	Синтаксис и операции. Типы данных языка и расширения типов, предоставляемые средой. Арифметические операции. Правила именования методов, переменных и констант. Основные алгоритмические конструкции. Использование структур, условных конструкций, циклов, функций, а также директив предпроцессора.	4	4	6	18
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. Наследование в ООП. Наследование, виды наследования. Абстрактные классы. Множественное наследование Управление данными, директивы предпроцессора. Структуры, исключения.	8	12	12	32
4	Разработка приложений с графическими интерфейсными возможностями	Введение в разработку пользовательского интерфейса. Понятие интерфейса пользователя, метафоры и поиск подобных решений, интерактивные средства разработки. Средства разработки интерфейса IDE сред. Основы разработки программных интерфейсов, средства разработки, иерархия и структура интерфейсных классов. Основы библиотеки Qt. Обработка событий в приложениях Qt. Средства взаимодействия между приложением и действиями, исходящими от пользователя, стандартные элементы управления. Однооконные и многооконные приложения. Обработка данных, запись результатов (SDI и MDI). Организация вывода графической информации. Работа с графикой на плоскости, технологии	8	18	12	34

		реализации вывода в 3D, подключение к проектам библиотек, поддерживающих OpenGL. Представление, ввод и вывод данных на носители информации. Работа с файлами, поддержка баз данных, создание архивов и извлечение информации из них, регулярные выражения.				
5	Инструментальные средства моделирования структур данных и алгоритмов	Основы UML. Диаграммы классов, прецедентов, последовательностей. Схемы баз данных. Инструментальные средства моделирования.	4	4	4	12
6	Способы тестирования приложений	Ручное и автоматическое тестирование. Пирамида тестирования. Создание плана тестирования. Средства разработки автоматических тестов на C++. Автоматическое тестирование графических интерфейсов приложений. Анализ покрытия кода.	4	8	12	24
7	Контроль качества программного кода	Статические анализаторы кода. Показатели сложности программного кода. Инструментальные средства контроля сложности.	4	4	4	12
Итого			36	54	54	144

заочная форма обучения

№ п/п	Наименование темы	Содержание раздела	Лекц	Лаб. зан.	СРС	Всего, час
1	Интегрированные среды и средства разработки на языке программирования высокого уровня	Введение. Анализ языков программирования и обзор сред разработки программного обеспечения. Возможности и производительность современных компиляторов, сборщиков проектов. Среды разработки. Интерфейс и конфигурация интегрированной среды разработки (IDE) под решение вычислительных задач. Разработка консольных проектов и приложений с графическим интерфейсом.	1	2	11	24
2	Базовые конструкции и ключевые слова языка программирования высокого уровня	Синтаксис и операции. Типы данных языка и расширения типов, предоставляемые средой. Арифметические операции. Правила именования методов, переменных и констант. Основные алгоритмические конструкции. Использование структур, условных конструкций, циклов, функций, а также директив предпроцессора.	-	2	17	20
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. Наследование в ООП. Наследование, виды наследования. Абстрактные классы. Множественное наследование Управление данными, директивы предпроцессора. Структуры, исключения.	-	-	38	38
4	Разработка приложений с графическими интерфейсными возможностями;	Введение в разработку пользовательского интерфейса. Понятие интерфейса пользователя, метафоры	1	-	39	40

		и поиск подобных решений, интерактивные средства разработки. Средства разработки интерфейса IDE сред. Основы разработки программных интерфейсов, средства разработки, иерархия и структура интерфейсных классов. Основы библиотеки Qt. Обработка событий в приложениях Qt. Средства взаимодействия между приложением и действиями, исходящими от пользователя, стандартные элементы управления. Однооконные и многооконные приложения. Обработка данных, запись результатов (SDI и MDI). Организация вывода графической информации. Работа с графикой на плоскости, технологии реализации вывода в 3D, подключение к проектам библиотек, поддерживающих OpenGL. Представление, ввод и вывод данных на носители информации. Работа с файлами, поддержка баз данных, создание архивов и извлечение информации из них, регулярные выражения.				
5	Инструментальные средства моделирования структур данных и алгоритмов	Основы UML. Диаграммы классов, прецедентов, последовательностей. Схемы баз данных. Инструментальные средства моделирования.	-	-	10	10
6	Способы тестирования приложений	Ручное и автоматическое тестирование. Пирамида тестирования. Создание плана тестирования. Средства разработки автоматических тестов на C++. Автоматическое тестирование графических интерфейсов приложений. Анализ покрытия кода.	-	-	38	38
7	Контроль качества программного кода	Статические анализаторы кода. Показатели сложности программного кода. Инструментальные средства контроля сложности.	-	-	11	11
Итого			2	4	165	171

5.2 Перечень лабораторных работ

1. Разработка и сборка простого консольного приложения.
2. Создание и наследование классов в C++.
3. Интерфейсы и абстрактные классы в C++.
4. Контейнеры и вложенные циклы в C++.
5. Чтение файлов в C++. Интерфейс командной строки.
6. Разработка GUI-приложений в Qt. Компоновка элементов, вывод сообщений.
7. Разработка GUI-приложений в Qt. Рисование в окне, обработка событий мыши.
8. UML-моделирование.
9. Подпроекты и автотесты.

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

В соответствии с учебным планом освоение дисциплины предусматривает выполнение курсового проекта в 3 семестре для очной формы обучения, в 3 семестре для заочной формы обучения.

Примерная тематика курсового проекта: «Разработка на C++/Qt приложения, обеспечивающего функционирование медицинского центра».

Задачи, решаемые при выполнении курсового проекта:

- Проектирование приложения;
- Реализация приложения;
- Тестирование приложения;
- Написание пояснительной записки.

Курсовой проект включают в себя графическую часть и расчетно-пояснительную записку.

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-8	Знать основные объекты представления данных (классы) и методы их обработки, а также способы реализации	Количество выполненных лабораторных работ	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	Уметь реализовывать алгоритмы на языке высокого уровня	Количество выполненных 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	Количество выполненных лабораторных работ	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
ОПК-9	Знать методы и технологии программирования	Количество выполненных 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	Количество выполненных лабораторных работ	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах

			программах	программах
	Владеть методами описания основных классов и алгоритмов на их основе	Количество выполненных 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 в 3 семестре для очной формы обучения, 3 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-8	Знать основные объекты представления данных (классы) и методы их обработки, а также способы реализации	Тест	Выполнение теста на 90- 100%	Выполнение теста на 80- 90%	Выполнение теста на 70- 80%	В тесте менее 70% правильных ответов
	Уметь реализовывать алгоритмы на языке высокого уровня	Решение стандартных практических задач	Задачи решены в полном объеме и получены верные ответы	Продемонстрирован верный ход решения всех, но не получен верный ответ во всех задачах	Продемонстрирован 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	Продемонстрирован верный ход решения всех, но не получен верный ответ во всех задачах	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
ОПК-9	Знать методы и технологии программирования	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	Решение стандартных практических задач	Задачи решены в полном объеме и получены верные ответы	Продемонстрирован верный ход решения всех, но не получен верный ответ во всех задачах	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
	Владеть методами описания основных классов и алгоритмов на их основе	Решение прикладных задач в конкретной	Задачи решены в полном объеме и	Продемонстрирован верный ход решения	Продемонстрирован верный ход решения в	Задачи не решены

		предметной области	получены верные ответы	всех, но не получен верный ответ во всех задачах	большинстве задач	
--	--	--------------------	------------------------	--	-------------------	--

7.2 Примерный перечень оценочных средств (типичные 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. Какому понятию ООП относится следующее определение: «Совокупность свойств (параметров) определенных сущностей и методов их обработки»:

- Свойство;
- + Объект;
- Метод;
- Событие;
- Класс.

2. Выберите типы данных C++, подходящих для хранения целочисленных значений:

- + char;
- bool;
- + int;
- + long;
- double.

3. Какой цикл является циклом с постусловием:

- for;
- + do while;
- while;
- goto.

4. С помощью какой зарезервированной процедуры выполняются заключительные действия по освобождению объекта в динамической памяти

- Constructor;
- + Destructor;
- Destroy;
- Create;
- Cancel

5. Установите правильный порядок действий, для того, чтобы изменить стандартное имя компонента QLabel в среде QtDesigner:

- 1 выделить элемент QLabel на форме
- 2 перейти в режим дизайна формы
- 3 выбрать вкладку Properties
- 4 изменить свойство Text для объекта QLabel

6. Укажите характеристики виртуального метода:

- выделяется время на выполнение, а затем предоставляется ресурс;

- создается таблица виртуальных методов;
- + определяется директивой virtual;
- создается таблица динамических методов.

7. Как в объектно-ориентированном программировании называется свойство объектов разных классов выполнять одно и то же действие по-своему, т.е. способность реагировать на запрос согласно своему типу:

- инкапсуляция;
- наследование;
- + полиморфизм;
- повторение;
- перегружаемость;
- для выполнения выделяется сначала место в памяти, а затем предоставляется время на выполнение.

8. Переменная X является указателем на некоторый объект, который освобождается вызовом delete. Чему равно значение X после выполнения этого кода?

- 0x0;
- Тому же, чему было равно до вызова delete;
- Произвольному случайному значению;
- Null;
- Равно 5.

9) Какая директива используется в C++ для того чтобы в теле метода класса обратиться к данному классу:

- + this;
- self;
- private;
- class;
- virtual.

10. Выберите те понятия, которые относятся к свойствам основных понятий ООП:

- + инкапсуляция;
- + свойство;
- + наследование;
- + полиморфизм;
- + объект.

7.2.2 Примерный перечень заданий для решения стандартных задач

1. Библиотека Qt - это ...

Ответ: 1

- (1) набор классов C++ для создания графических приложений
- (2) язык разработки графических интерфейсов
- (3) язык программирования
- (4) язык манипулирования данными

2. Модуль QtCore содержит ...

Ответ: 2

(1) классы Qt для работы с документами

(2) ядро функциональности Qt

(3) логику приложения

(4) бизнес-логику приложения

3. Модуль QtGui содержит ...

Ответ: 3

(1) ядро функциональности Qt

(2) классы Qt для работы с документами

(3) классы Qt для работы с событиями и графикой

(4) стандартные графические элементы

4. Модуль QtWidgets содержит ...

Ответ: 4

(1) ядро функциональности Qt

(2) классы Qt для работы с документами

(3) классы Qt для работы с событиями и графикой

(4) стандартные графические элементы

5. Класс ... обеспечивает функционирование сигналов/слотов в Qt.

Ответ: 2

(1) QWidget

(2) QObject

(3) QtCore

(4) QSignal

6. Соединение сигнала и слота обеспечивается функцией ...

Ответ: 1

(1) connect

(2) attach

(3) handle

(4) collaborate

7. Добавление метаданных в производный класс от QObject обеспечивается макросом ...

Ответ: 1

(1) Q_OBJECT

(2) Q_META

(3) Q_ADD_META

(4) Q_CLASS_INFO

8. Фильтрация событий, поступающих в QObject, выполняется при переопределении метода QObject:: ...

Ответ: 1

(1) eventFilter()

(2) filter()

(3) event()

(4) handle()

9. Создание иерархических структур экземпляров QObject позволяет ...

Ответ: 2

- (1) реализовать последовательный рендер
- (2) организовать каскадное удаление объектов
- (3) выводить сообщения

10. Очередь обработки событий в Qt запускается методом:

Ответ: 1

- (1) exec()
- (2) run()
- (3) perform()
- (4) eventQueue()

7.2.3 Примерный перечень заданий для решения прикладных задач

1. Вывод статического текста в приложении Qt обеспечивается виджетом:

Ответ: 1

- (1) QLabel
- (2) QStatic
- (3) QTextEdit
- (4) QLineEdit

2. Ввод целочисленных значений в приложении Qt обеспечивается виджетом:

Ответ: 2

- (1) QLabel
- (2) QSpinBox
- (3) QTextEdit
- (4) QLineEdit

3. Ввод чисел с плавающей точкой в приложении Qt обеспечивается виджетом:

Ответ: 3

- (1) QLabel
- (2) QSpinBox
- (3) QDoubleSpinBox
- (4) QLineEdit

4. Ввод целочисленных значений в приложении Qt обеспечивается виджетом:

Ответ: 2

- (1) QLabel
- (2) QSlider
- (3) QTextEdit
- (4) QLineEdit

5. Ввод однострочного текста в приложении Qt обеспечивается виджетом:

Ответ: 4

- (1) QLabel
- (2) QSpinBox

(3) QDoubleSpinBox

(4) QLineEdit

6. Ввод многострочного текста в приложении Qt обеспечивается виджетом:

Ответ: 4

(1) QLabel

(2) QSpinBox

(3) QDoubleSpinBox

(4) QTextEdit

7. Переключатель вкл/выкл в приложении Qt обеспечивается виджетом:

Ответ: 2

(1) QLabel

(2) QCheckBox

(3) QDoubleSpinBox

(4) QComboBox

8. Выбор «один из многих» в приложении Qt обеспечивается виджетом:

Ответ: 4

(1) QLabel

(2) QCheckBox

(3) QDoubleSpinBox

(4) QComboBox

9. Представление таблиц в приложении Qt обеспечивается виджетом:

Ответ: 2

(1) QLabel

(2) QTableWidgetItem

(3) QDoubleSpinBox

(4) QComboBox

10. Представление древовидных структур в приложении Qt обеспечивается виджетом:

Ответ: 1

(1) QTreeWidget

(2) QCheckBox

(3) QDoubleSpinBox

(4) QComboBox

7.2.4 Примерный перечень вопросов для подготовки к зачету

Не предусмотрено учебным планом.

7.2.5 Примерный перечень заданий для подготовки к экзамену

1. Компилятор - определение, принцип работы, этапы компиляции, входные и выходные данные. Компиляторы C++. Компоновщик C++.

2. Makefile - определение, назначение, структура, построение зависимостей. Программы для выполнения сценария Makefile.

3. Структура исполняемого файла .exe (Portable Executable Win32).
4. Системы автоматизации сборки в C++: cmake и qmake. Особенности, принципы работы.
5. Интегрированные среды разработки на C++. QtCreator. Основные экраны в Qt Creator: редактор, отладчик, мастер создания, настройки сборки и т.д.
6. Концепции объектно-ориентированного программирования (“три кита ООП”). Определения класса и объекта. Конструктор и деструктор класса в C++. Указатель “this”.
7. Члены класса, контроль доступа к ним в C++. Использование слов “static” и “const” для членов класса. Отличия структур от классов в C++.
8. Наследование в объектно-ориентированном программировании на C++: базовый класс и производный класс, спецификаторы доступа при наследовании, множественное наследование.
9. Компьютерные пользовательские интерфейсы: типы, характеристики, отличия, применимость.
10. Интерфейс командной строки: определение, типы аргументов, аргумент “help”, реализация интерфейса командной строки в C++.
11. Библиотека Qt: основные модули, объектная модель Qt (класс QObject, сигналы/слоты, иерархия объектов).
12. Графический пользовательский интерфейс в Qt: класс QWidget, обработка событий в классе QWidget (клавиатуры, мыши и т.д.), компоновщики виджетов.
13. Стандартные элементы управления в Qt: кнопки, флажки, переключатели, ввод целочисленных значений и значений с плавающей точкой, ввод однострочного и многострочного текста, выбор “один из многих”.
14. Стандартные элементы управления в Qt: меню, статический текст, полосы прокрутки, индикация прогресса, списки, таблицы, древовидные структуры.
15. Средства проектирования графических пользовательских интерфейсов: Figma, diagrams.net, QtDesigner и другие.
16. Диаграммы UML: диаграмма прецедентов, схема базы данных, блок-схема алгоритма.
17. Диаграммы UML: схема классов, диаграмма последовательности.
18. Инструментальные средства построения диаграмм и их особенности: diagrams.net, StarUML, PlantUML и другие.
19. Автоматизация тестирования приложений: пирамида тестирования, юнит-тесты.
20. Юнит-тесты в Qt: модуль Qt Test, структура юнит-теста в Qt Test.
21. Статический анализ кода: определение, какие проблемы выявляет. Примеры статических анализаторов в C++.
22. Сложность программ, какие программы и проекты называем сложными. Количественные характеристики (метрики) программ. Инструменты для вычисления метрик кода C++.

23. Системы контроля версий: определение, назначение. Особенности системы контроля версий git, ее отличия от других аналогичных систем.

24. Ветки и теги в git: определение, назначение. Пример рабочего процесса в git (git workflow).

25. Анализ покрытия кода: определение, назначение, показатели. Примеры анализаторов покрытия кода в C++.

7.2.6. Методика выставления оценки при проведении промежуточной аттестации

Экзамен проводится по тест-билетам, каждый из которых содержит 10 вопросов и задачу. Каждый правильный ответ на вопрос в тесте оценивается 1 баллом, задача оценивается в 10 баллов (5 баллов верное решение и 5 баллов за верный ответ). Максимальное количество набранных баллов – 20.

1. Оценка «Неудовлетворительно» ставится в случае, если студент набрал менее 6 баллов.

2. Оценка «Удовлетворительно» ставится в случае, если студент набрал от 6 до 10 баллов

3. Оценка «Хорошо» ставится в случае, если студент набрал от 11 до 15 баллов.

4. Оценка «Отлично» ставится, если студент набрал от 16 до 20 баллов.

7.2.7 Паспорт оценочных материалов

№ п/п	Контролируемые разделы (темы) дисциплины	Код контролируемой компетенции	Наименование оценочного средства
1	Интегрированные среды и средства разработки на языке программирования высокого уровня	ОПК-8, ОПК-9	Тест, , защита лабораторных работ, требования к курсовому проекту
2	Базовые конструкции и ключевые слова языка программирования высокого уровня	ОПК-8, ОПК-9	Тест, , защита лабораторных работ, требования к курсовому проекту
3	Объектно-ориентированный подход к разработке программных средств	ОПК-8, ОПК-9	Тест, , защита лабораторных работ, требования к курсовому проекту
4	Разработка приложений с графическими интерфейсными возможностями	ОПК-8, ОПК-9	Тест, , защита лабораторных работ, требования к курсовому проекту
5	Инструментальные средства моделирования структур данных и алгоритмов	ОПК-8, ОПК-9	Тест, , защита лабораторных работ, требования к курсовому проекту
6	Способы тестирования приложений	ОПК-8, ОПК-9	Тест, , защита лабораторных работ, требования к курсовому проекту

7	Контроль качества программного кода	ОПК-8, ОПК-9	Тест, , защита лабораторных работ, требования к курсовому проекту
---	-------------------------------------	--------------	---

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 30 мин. Затем осуществляется проверка теста экзаменатором и выставляется оценка согласно методике выставления оценки при проведении промежуточной аттестации.

Решение стандартных задач осуществляется, либо при помощи компьютерной системы тестирования, либо с использованием выданных задач на бумажном носителе. Время решения задач 30 мин. Затем осуществляется проверка решения задач экзаменатором и выставляется оценка, согласно методике выставления оценки при проведении промежуточной аттестации.

Решение прикладных задач осуществляется, либо при помощи компьютерной системы тестирования, либо с использованием выданных задач на бумажном носителе. Время решения задач 30 мин. Затем осуществляется проверка решения задач экзаменатором и выставляется оценка, согласно методике выставления оценки при проведении промежуточной аттестации.

Защита курсовой работы, курсового проекта или отчета по всем видам практик 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

1. Шилдт, Г. Самоучитель С++ / Г. Шилдт. — 3-е изд. — СПб : БХВ-Петербург, 2005. — 688 с. — ISBN 5-7791-0086-1.

2. Шлее, М. Qt 5.10. Профессиональное программирование на С++ / М. Шлее. — СПб : БХВ-Петербург, 2018. — 1072 с. — ISBN 978-5-9775-3678-3.

3. Саттер, Герб, Александреску, Андрей. Стандарты программирования на С++. — М : Издательский дом «Вильямс», 2005. — 224 с. — ISBN 5-8459-0859-0.

4. Юров А.Н., Методические указания к лабораторным работам № 1-3 по дисциплине “Программирование” 14-2013, 2013.

5. Юров А.Н., Методические указания к лабораторным работам № 4-5 по дисциплине “Программирование” 15-2013, 2013.

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:

Лицензионное ПО:

- Microsoft Word

Свободное программное обеспечение:

- MS Visual Studio Community Edition

- LibreOffice

Отечественное ПО:

- СУБД Линтер

Ресурсы информационно-телекоммуникационной сети «Интернет»:

- <http://www.edu.ru/>

- Образовательный портал ВГТУ

Информационные справочные системы:

- <http://window.edu.ru>

- <https://wiki.cchgeu.ru/>

9 МАТЕРИАЛЬНО-ТЕХНИЧЕСКАЯ БАЗА, НЕОБХОДИМАЯ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА

Специализированная лекционная аудитория, оснащенная оборудованием для лекционных демонстраций и проекционной аппаратурой.

Учебные лаборатории (г. Воронеж, ул. Плехановская, д. 11):

- “Компьютерное моделирование и дизайн”.

- “Интеллектуальные системы проектирования”.

Дисплейный класс, оснащенный компьютерными программами для проведения лабораторного практикума.

Кабинеты, оборудованные проекторами и интерактивными досками.

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

По дисциплине «Среды визуального программирования» 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; пометать 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: <ul style="list-style-type: none"> - работа с текстами: учебниками, справочниками, дополнительной литературой, а также проработка конспектов лекций; - выполнение домашних заданий и расчетов; - работа над темами для самостоятельного изучения; - курсовое проектирование; - 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 начаться не позднее, чем за месяц-полтора до промежуточной аттестации. Данные перед экзаменом, экзаменом три дня эффективнее всего использовать для повторения и систематизации материала.