


**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 /
И.О. Фамилия
«31» августа 2021 г.

**РАБОЧАЯ ПРОГРАММА
дисциплины**

«Среды визуального программирования»

Направление подготовки 09.03.01 Информатика и вычислительная техника

код и наименование направления подготовки/специальности

Профиль Программное обеспечение автоматизированных систем

название профиля/программы

Квалификация выпускника бакалавр

Нормативный период обучения 4 г. / 4 г. 11 мес.

Очная/очно-заочная/заочная (при наличии)

Форма обучения Очная/Заочная

Год начала подготовки 2021

Автор программы



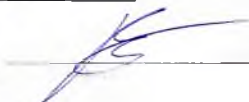
/В.А. Рыжков/

Заведующий кафедрой
Компьютерных
интеллектуальных
технологий проектирования



/М.И. Чижев/

Руководитель ОПОП



В.В. Ветехин

Воронеж 2021

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)	14	14
В том числе:		
Лекции	2	2
Лабораторные работы (ЛР)	12	12
Самостоятельная работа	157	157
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	8	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, набор директив для сборки проектов	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 Обработка событий в приложениях Средства взаимодействия между приложением и действиями, исходящими от пользователя Виды приложений и особенности их создания Однооконные и многооконные приложения. Обработка данных, запись результатов (SDI и MDI) Организация вывода графической информации Работа с графикой на плоскости, технологии	4	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	Создание подсистем на основе подключаемых библиотек;	Динамически-подключаемые источники данных Работа с системными библиотеками операционных систем	4	4	4	12
6	Построение кроссплатформенных приложений;	Введение и особенности создания межплатформенных проектов Построение кроссплатформенных приложений средствами IDE Технологии и инструменты сред разработки по созданию кроссплатформенных приложений Использование базовых классов среды. Перенос данных.	4	8	12	24
7	Создание инсталляционных пакетов программ.	Создание установочных пакетов Средства и решения для автоматического развёртывания проектов на ЭВМ пользователя	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 препроцессора	1	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 препроцессора Структуры, исключения, набор директив для сборки проектов	-	1	37	38

4	Разработка приложений с графическими интерфейсными возможностями;	<p>Введение в разработку пользовательского интерфейса</p> <p>Понятие интерфейса пользователя, метафоры и поиск подобных решений, интерактивные средства разработки</p> <p>Средства разработки интерфейса IDE сред</p> <p>Основы разработки программных интерфейсов, средства разработки, иерархия и структура интерфейсных классов</p> <p>Обработка событий в приложениях</p> <p>Средства взаимодействия между приложением и действиями, исходящими от пользователя</p> <p>Виды приложений и особенности их создания</p> <p>Однооконные и многооконные приложения. Обработка данных, запись результатов (SDI и MDI)</p> <p>Организация вывода графической информации</p> <p>Работа с графикой на плоскости, технологии реализации вывода в 3D, подключение к проектам библиотек, поддерживающих OpenGL</p> <p>Представление, ввод и вывод данных на носители информации</p> <p>Работа с файлами, поддержка баз данных, создание архивов и извлечение информации из них, регулярные выражения</p>	1	4	36	40
5	Создание подсистем на основе подключаемых библиотек;	<p>Динамически-подключаемые источники данных</p> <p>Работа с системными библиотеками операционных систем</p>	-	1	9	10
6	Построение кроссплатформенных приложений;	<p>Введение и особенности создания межплатформенных проектов</p> <p>Построение кроссплатформенных приложений средствами IDE</p> <p>Технологии и инструменты сред разработки по созданию кроссплатформенных приложений</p> <p>Использование базовых классов среды. Перенос данных.</p>	-	1	37	38
7	Создание установочных пакетов программ.	<p>Создание установочных пакетов</p> <p>Средства и решения для автоматического развёртывания проектов на ЭВМ пользователя</p>	1	1	9	11
Итого			2	12	157	171

5.2 Перечень лабораторных работ

1. Обзор возможностей среды разработки Microsoft Visual Studio 2019
2. Знакомство с технологией Windows Presentation Foundation
3. Создание динамически подключаемых библиотек.
4. Знакомство с технологиями совместной разработки ПО. Система контроля версий GitHub.
5. Интернационализация Windows Presentation Foundation приложений
6. Прототипирование приложений
7. Отладка приложений в Visual Studio. Изучение возможностей отладчика.
8. Тестирование приложений в MS Visual Studio 2019. Изучение

возможностей профилировщика.

9. Коллекции C#. Работа с LINQ.

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 приложения обеспечивающего функционирование медицинского центра в MS Visual Studio».

Задачи, решаемые при выполнении курсового проекта:

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

Курсовой проект включает 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
	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	Решение прикладных	Задачи решены в	Продемонстрирован	Продемонстрирован	Задачи не решены

	и алгоритмов на их основе	задач в конкретной предметной области	в полном объеме и получены верные ответы	верный ход решения всех, но не получен верный ответ во всех задачах	верный ход решения в 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 те обозначения имен полей, которые допустимы в среде MS Visual Studio

- + I:integer;
- _:real;
- + _2: integer;
- + _n:real;
- 2:integer.

3. Какая директива используется для восстановления обработок ошибок ввода/вывода

- {\$R *.res};
- + {\$I -};
- {\$F+};
- {\$I +};
- {\$R *.dfm}

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 Label, как в приведенном отрывке модуля

- 1 перейти в окно инспектора объектов
- 2 поместить на форму компонент Label
- 3 выбрать вкладку Properties
- 4 изменить свойство Name для объекта Label

6. Укажите характеристики динамического 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 ссылкой на некоторый объект, который освобождается вызовом `X.Free`. Чему равно значение `X` после выполнения этого кода?

- `nil`
- Тому же, чему было равно до вызова `Free`
- Произвольному случайному значению
- Произвольному случайному значению, но обязательно отличному от

`nil`

- Равно 5

9) Какая директива используется для того чтобы в теле перекрывающего метода обратиться к методу предка

- + `inherited`
- `public`
- `private`
- `abstract`
- `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. Язык XAML это ...

Ответ: 1

- (1) язык расширенной разметки приложений
- (2) язык расширенной разметки
- (3) язык программирования
- (4) язык манипулирования данными

2. XAML-документ содержит ...

Ответ: 1 и 2

(1) разметку, описывающую внешний вид и поведение окна приложения

(2) разметку, описывающую внешний вид и поведение страницы приложения

(3) логику приложения

(4) бизнес-логику приложения

3. Укажите правильные утверждения

Ответ: 1

(1) WPF XAML является подмножеством языка XML

(2) XML является подмножеством языка WPF XAML

(3) XAML является подмножеством языка C#

(4) C# является подмножеством языка XAML

4. Укажите правильные утверждения в отношении синтаксиса XAML

Ответ: 1, 2, 3

(1) элемент XAML-документа отображается на некоторый экземпляр класса .NET

(2) элементы XAML можно вкладывать друг в друга

(3) свойства класса определяются с помощью атрибутов

(4) свойства класса определяют бизнес-логику

5. Все дескрипторы XAML-документа начинаются символом ...

Ответ: 2

(1) >

(2) <

(3) &

(4) :

6. Все дескрипторы XAML-документа завершаются символом ...

Ответ: 1

(1) >

(2) <

(3) &

(4) :

7. Текст XAML-документа должен содержать ...

Ответ: 1

(1) один корневой элемент

(2) два корневых элемента

(3) любое количество корневых элементов

(4) только корневые элементы

8. Пространство имен в XAML-документе задается с помощью атрибута

Ответ: 1

(1) xmlns

(2) http

(3) window

(4) page

9. Соответствие между ключевым словом XAML и его назначением

Ответ: 2

(1)

x:Array Представляет тип массива .NET на XAML

x:FieldModifier Позволяет определять видимость типа класс

x:ClassModifier Позволяет определять видимость члена типа

x:Key Позволяет установить значение ключа для элемента XAML

(2)

x:Array Представляет тип массива .NET на XAML

x:ClassModifier Позволяет определять видимость типа класс

x:FieldModifier Позволяет определять видимость члена типа

x:Key Позволяет установить значение ключа для элемента XAML

(3)

x:Key Представляет тип массива .NET на XAML

x:ClassModifier Позволяет определять видимость типа класс

x:FieldModifier Позволяет определять видимость члена типа

x:Array Позволяет установить значение ключа для элемента XAML

10. Соответствие между ключевым словом XAML и его назначением

Ответ: 1

(1)

x:Name Позволяет указывать сгенерированное C# имя заданного элемента XAML

x:Null Представляет null-ссылку

x:Static Позволяет ссылаться на статический член типа

x:TypeArgument Позволяет устанавливать элемент как обобщенный тип с определенными параметрами

(2)

x:TypeArgument Позволяет указывать сгенерированное C# имя заданного элемента XAML

x:Null Представляет null-ссылку

x:Static Позволяет ссылаться на статический член типа

x:Name Позволяет устанавливать элемент как обобщенный тип с определенными параметрами

(3)

x:Name Позволяет указывать сгенерированное C# имя заданного элемента XAML

x:Null Представляет null-ссылку

x:TypeArgument Позволяет ссылаться на статический член типа

x:Static Позволяет устанавливать элемент как обобщенный тип с определенными параметрами

7.2.3 Примерный перечень заданий для решения прикладных задач

1 Пространство

имен:

<http://schemas.microsoft.com/winfx/2006/xaml/presentation>

Ответ: 1

(1) базовое пространство имен WPF, которое охватывает все классы WPF

(2) пространство имен XAML, которое включает различные свойства утилит XAML

(3) пространство имен XAML, связанное с проблемой совместимости разметки с рабочей средой

(4) пространство имен XAML, поддерживаемое программами Expression Blend и Visual Studio

2 Пространство имен: <http://schemas.microsoft.com/winfx/2006/xaml>

Ответ: 2

(1) базовое пространство имен WPF, которое охватывает все классы WPF

(2) пространство имен XAML, которое включает различные свойства утилит XAML

(3) пространство имен XAML, связанное с проблемой совместимости разметки с рабочей средой

(4) пространство имен XAML, поддерживаемое программами Expression Blend и Visual Studio

3 Пространство имен: <http://schemas.openxmlformats.org/markup-compatibility/2006>

Ответ: 3

(1) базовое пространство имен WPF, которое охватывает все классы WPF

(2) пространство имен XAML, которое включает различные свойства утилит XAML

(3) пространство имен XAML, связанное с проблемой совместимости разметки с рабочей средой

(4) пространство имен XAML, поддерживаемое программами Expression Blend и Visual Studio

4 Пространство имен: <http://schemas.microsoft.com/expression/blend/2008>

Ответ: 4

(1) базовое пространство имен WPF, которое охватывает все классы WPF

(2) пространство имен XAML, которое включает различные свойства утилит XAML

(3) пространство имен XAML, связанное с проблемой совместимости разметки с рабочей средой

(4) пространство имен XAML, поддерживаемое программами Expression Blend и Visual Studio

5 Строка XAML-документа

```
<Window x:Class="MyProject.MainWindow" ...>
```

определяет ...

Ответ: 1

(1) создание дочернего класса Window – MyProject.MainWindow

(2) префикс x атрибута Class определяет то, что данный атрибут помещается в пространство имен <http://schemas.microsoft.com/expression/blend/2008>

(3) префикс x атрибута Class определяет то, что данный атрибут помещается в пространство имен <http://schemas.openxmlformats.org/markup-compatibility/2006>

6 В следующем XAML- описании

```
<Button Name="Exit" Content="Выход" />
```

фрагмент Name="Exit" определяет

Ответ: 1

- (1) простое свойство
- (2) сложное свойство
- (3) присоединенное свойство
- (4) расширение разметки

7 Дескриптор <StackPanel.Background> определяет

Ответ: 2

- (1) простое свойство
- (2) сложное свойство
- (3) присоединенное свойство
- (4) расширение разметки

8 В следующем XAML- описании

```
<Button Foreground="{x:Static SystemColors.ActiveCaptionBrush}" />
```

фрагмент Foreground="{x:Static SystemColors.ActiveCaptionBrush}"

определяет

Ответ: 4

- (1) простое свойство
- (2) сложное свойство
- (3) присоединенное свойство
- (4) расширение разметки

9 В следующем XAML- описании

```
<Button ... Grid.Row="0" ... />
```

фрагмент Grid.Row="0" определяет

Ответ: 3

- (1) простое свойство
- (2) сложное свойство
- (3) присоединенное свойство
- (4) расширение разметки

10 Номер 2

В следующем XAML- описании

```
<Button Name="Exit" Content="Выход" Click="Exit_Click" />
```

фрагмент Click="Exit_Click" определяет

Ответ: 4

- (1) простое свойство
- (2) сложное свойство
- (3) присоединенное свойство

(4) присоединение обработчика события

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. Среда .Net. Особенности разработки программ на C#.
2. Использование пространства имен (using namespace) в C#.
3. Базовые типы данных, различия в C++ и C#.
4. Объекты и классы в C#. Использование конструкторов. Наследование.

5. Обработка исключений. Примеры использования.

6. Модификаторы ref и out. Применение конструкций set и get для задания объектам начальных значений.

7. Разработка приложений с пользовательским интерфейсом на C#.

8. Обработка событий от элементов управления приложения.

9. Графический вывод объектов в окне приложения на C#.

10. Элемент табличного представления данных DataGridView.

11. Работа с динамическими библиотеками .dll.

12. Разработка приложений с поддержкой БД на C#.

13. Работа с файлами на C#.

14. Работа с клавиатурой и мышью в графическом приложении.

15. Параметризация пользовательского интерфейса приложения.

16. Поддержка графического вывода на основе OpenGL в C#.

17. Многопоточность в C#.

18. Регулярные выражения на C#.

19. Информационные сообщения в графическом приложении.

20. Стандартные пользовательские диалоги на 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	Создание подсистем на основе подключаемых библиотек;	ОПК-8, ОПК-9	Тест, , защита лабораторных работ, требования к курсовому проекту
6	Построение кроссплатформенных приложений;	ОПК-8, ОПК-9	Тест, , защита лабораторных работ, требования к курсовому проекту
7	Создание инсталляционных пакетов программ.	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

Кариев, Ч. А. Разработка Windows-приложений на основе Visual C# : учебное пособие / Ч. А. Кариев. — 3-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. — 978 с. — ISBN 978-5-4497-0909-7. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/102057.html>

2. Ганцева Е.А. Программирование на C#. Воронеж: ФГБОУ ВПО "Воронежский государственный технический университет", 2012. - 135с.

3. Разработка Windows-приложений в среде программирования Visual Studio.Net : учебно-методическое пособие по дисциплине Информатика и программирование / составители Ю. А. Воронцов, А. Г. Ерохин. — Москва : Московский технический университет связи и информатики, 2016. — 20 с. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/61536.html>

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; 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;

	<ul style="list-style-type: none"> - выполнение домашних заданий и расчетов; - работа над темами для самостоятельного изучения; - курсовое проектирование; - участие в работе студенческих научных конференций, олимпиад; - подготовка к промежуточной аттестации.
<p>Подготовка к промежуточной аттестации</p>	<p>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.</p>

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

№ п/п	Перечень вносимых изменений	Дата внесения изменений	Подпись заведующего кафедрой, ответственной за реализацию ОПОП
----------	-----------------------------	-------------------------------	--