

Документ подписан квалифицированной электронной подписью

Сертификат: 029405EA0079B1609A4E443133CFE7DA

Владелец: "АНО ВО "РОССИЙСКИЙ НОВЫЙ УНИВЕРСИТЕТ"; АИ

Действителен до: 2024-01-12 12:00:00

**АНО ВО «Российский новый университет»**

**Елецкий филиал Автономной некоммерческой организации высшего образования «Российский новый университет»  
(Елецкий филиал АНО ВО «Российский новый университет»)**

кафедра прикладной экономики

**Рабочая программа учебной дисциплины (модуля)**

Визуальное программирование  
(наименование учебной дисциплины (модуля))

09.03.03 Прикладная информатика  
(код и направление подготовки/специальности)

Прикладная информатика в экономике  
(код и направление подготовки/специальности, в случаях, если программа разработана для разных направлений подготовки/специальностей)

---

Рабочая программа учебной дисциплины (модуля) рассмотрена и утверждена на заседании кафедры «16» января 2024, протокол № 5.

Заведующий кафедрой Прикладной экономики  
(название кафедры)

к.э.н., доцент Преснякова Д.В.

(ученая степень, ученое звание, фамилия и инициалы, подпись заведующего кафедрой)

Елец  
2024 год

## **1. НАИМЕНОВАНИЕ И ЦЕЛЬ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)**

Учебная дисциплина «Визуальное программирование» изучается обучающимися, осваивающими образовательную программу «Прикладная информатика в экономике» в соответствии с Федеральным государственным образовательным стандартом высшего образования по направлению 09.03.03 Прикладная информатика (уровень бакалавриата), утвержденным приказом Министерства образования и науки РФ от 19.09.2017 N 922 (ФГОС ВО 3++).

Цель изучения дисциплины. В результате изучения курса студенты должны приобрести знания, умения и навыки, необходимые для создания приложений в интегрированной среде визуального программирования с использованием визуальных компонентов, компонентов баз данных, приложения для решения прикладных задач, в том числе приложения для управления базами данных, выполнять обработку ошибок, разрабатывать и тестировать приложения; формировании осознанного отношения личности к процессам самообразования и самосовершенствования; создание условий направленных на развитие личности для самоопределения и социализации обучающихся на основе социокультурных, духовно-нравственных ценностей и принятых в российском обществе правил и норм поведения в интересах человека, семьи, общества и государства.

Изучение учебной дисциплины направлено на подготовку обучающихся к осуществлению деятельности по концептуальному, функциональному и логическому проектированию систем среднего и крупного масштаба и сложности, планированию разработки или восстановления требований к системе, анализу проблемной ситуации заинтересованных лиц, разработке бизнес-требований заинтересованных лиц, постановки целей создания системы, разработки концепции системы и технического задания на систему, организации оценки соответствия требованиям существующих систем и их аналогов, представлению концепции, технического задания на систему и изменений в них заинтересованным лицам, организации согласования требований к системе, разработке шаблонов документов требований, постановке задачи на разработку требований к подсистемам и контроль их качества, сопровождению приемочных испытаний и ввода в эксплуатацию системы, обработке запросов на изменение требований к системе, определенных профессиональным стандартом «Системный аналитик», утвержденного приказом Министерства труда и социальной защиты Российской Федерации от 27.04.2023 № 367н.

## **2. МЕСТО ДИСЦИПЛИНЫ (МОДУЛЯ) В СТРУКТУРЕ ОП**

Учебная дисциплина Визуальное программирование относится к части учебного плана формируемой участниками образовательных отношений и изучается на 1 курсе.

2.1. Требования к предварительной подготовке обучающегося:

Материал дисциплины «Визуальное программирование» базируется на знаниях, полученных в результате изучения дисциплины «Информатика и программирование». Параллельно с учебной дисциплиной «Визуальное программирование» изучаются дисциплины: «Информатика и программирование», «Современные языки и среды программирования».

2.2. Дисциплины (модули) и практики, для которых освоение данной дисциплины (модуля) необходимо как предшествующее:

Знания, полученные в ходе изучения дисциплины, понадобятся для освоения следующих курсов: «Программная инженерия», «Проектирование информационных систем».

Развитие у обучающихся навыков командной работы, межличностной коммуникации, принятия решений, лидерских качеств обеспечивается чтением лекций, проведением практических занятий, нацеленных на профессиональную деятельность выпускников и потребности работодателей.

### 3. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ, СООТНЕСЕННЫЕ С ПЛАНИРУЕМЫМИ РЕЗУЛЬТАТАМИ ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

В результате освоения дисциплины обучающийся должен овладеть следующими компетенциями:

- *Способен проектировать информационные системы по видам обеспечения (ПК-3)*

#### Планируемые результаты обучения по дисциплине.

Формируемая компетенция	Планируемые результаты обучения	Код результата обучения
Способен проектировать информационные системы по видам обеспечения (ПК-3)	<b><u>Знать:</u></b>	
	- концепции визуального программирования.	ПК-3-31
	- задавать свойства, отвечающие за визуальное представление формы на экране.	ПК-3-32
	- компоненты приложения, предназначенные для работы с базами данных.	ПК-3-33
	- способы компоновки элементов управления.	ПК-3-34
	<b><u>Уметь</u></b>	
	- задавать свойства, отвечающие за визуальное представление формы на экране.	ПК-3-У1
	- задавать свойства формы, отвечающие за ее поведение во время выполнения приложения.	ПК-3-У2
	- группировать и размещать элементы управления с целью создания удобного пользовательского интерфейса.	ПК-3-У3
	- определять оптимальные формы представления и адаптировать их с учетом уровня подготовленности коллег по совместной работе	ПК-3-У4
	<b><u>Владеть</u></b>	
	- разработки дружелюбного интерфейса пользователя.	ПК-3-В1
	- разработки прикладного программного обеспечения.	ПК-3-В2
	- внедрения и адаптации прикладного программного обеспечения.	ПК-3-В3
- определять инструментарий, необходимый для соответствующего анализа при автоматизации решения прикладных задач и создания ИС.	ПК-3-В4	

### 4. ОБЪЕМ ДИСЦИПЛИНЫ (МОДУЛЯ) В ЗАЧЕТНЫХ ЕДИНИЦАХ С УКАЗАНИЕМ КОЛИЧЕСТВА АКАДЕМИЧЕСКИХ ЧАСОВ, ВЫДЕЛЕННЫХ НА КОНТАКТНУЮ РАБОТУ

Общая трудоемкость дисциплины составляет 2 зачетных единицы (72 часа).

№	Форма обучения	Семестр/сессия, курс	Общая трудоемкость		в том числе контактная работа с преподавателем							СР	Контроль
			в з.е.	в часах	Всего	Л	ПР	КоР	зачет	Конс	экзамен		
1.	Заочная	Установочная сессия, 1 курс		36	4	4						32	

		Зимняя сессия, 1 курс	2	36	6		4	1,7	0,3			26,3	3,7
		Итого:	2	72	10	4	4	1,7	0,3			58,3	3,7

## Распределение учебного времени по темам и видам учебных занятий

### заочная форма обучения

№ №	Наименование разделов, тем учебных занятий	Всего часов	Контактная работа с преподавателем							СР	Формируемые результаты обучения	
			Всего	Л	Пр	Коп	Зачет	Конс	Экзамен			
<b>Формы Windows и пользовательский интерфейс</b>												
1.	Формы Windows и пользовательский интерфейс	9	1	1						8	ПК-3-31 ПК-3-32 ПК-3-33	
<b>Компоновка элементов управления</b>												
2.	Компоновка элементов управления	9	1	1						8	ПК-3-У2 ПК-3-В1 ПК-3-34	
<b>Управляющие элементы, обеспечивающие выбор из фиксированного набора альтернатив, предоставленных пользователю</b>												
3.	Управляющие элементы, обеспечивающие выбор из фиксированного набора альтернатив, предоставленных пользователю	9	1		1					8	ПК-3-32 ПК-3-У3 ПК-3-У4	
<b>Создание и использование многоуровневого меню</b>												
4.	Создание и использование многоуровневого меню	10	2	1	1					8	ПК-3-У1 ПК-3-У2 ПК-3-В1	
<b>Отображение информации о состоянии программы и визуализация выполняемых операций</b>												
5.	Отображение информации о состоянии программы и визуализация выполняемых операций	9	1	1						8	ПК-3-34 ПК-3-У3 ПК-3-У4	
<b>Компоненты, предназначенные для работы с базами данных</b>												
6.	Компоненты, предназначенные для работы с базами данных	9	1		1					8	ПК-3-33 ПК-3-В4	
<b>Создание ресурсов приложения</b>												
7.	Создание ресурсов приложения	11,3	1		1					10,3	ПК-3-В2 ПК-3-В3	
<b>Промежуточная аттестация (экзамен)</b>												
8.	Промежуточная аттестация (экзамен)	2	2			1,7	0,3					

## **5. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ), СТРУКТУРИРОВАННОЕ ПО ТЕМАМ (РАЗДЕЛАМ)**

### **Тема 1. Формы Windows и пользовательский интерфейс.**

Основы работы с интегрированной средой разработки Microsoft Visual Studio. Формы. Обзор форм Windows. Создание проекта. Режимы дизайнера и кода. Размещение элементов управления на форме. Компиляция и выполнение проекта. Окно проводника проекта. Компоненты, входящие в состав проекта. Свойства проекта. События в Windows – приложениях. События мыши и клавиатуры. Форматирование элементов управления. Окно Свойства. Получение доступа к методам и свойствам формы. Добавление формы в приложение во время выполнения. Изменение внешнего вида и поведения формы. Настройка состояния формы при запуске. Настройка стартовой формы. Окно свойств проекта. Создание нестандартных форм Windows.

### **Тема 2. Компоновка элементов управления.**

Обзор контейнерных элементов управления. Коллекция Controls. Добавление в конструкторе элемента управления в форму или контейнерный элемент управления. Добавление элемента управления в режиме работы приложения. Удаление заданного элемента в режиме работы приложения. Свойства элемента управления, определяющие его поведение внутри формы или родительского элемента управления. Группировка элементов управления. Элемент управления GroupBox. Создание самостоятельных подразделов формы. Элемент управления Panel и FlowLayoutPanel. Создание разрыва заливки на элементе. Использование табличных контейнеров. Группировка элементов управления на отдельных вкладках. Элемент управления TabControl. Создание подраздела формы. Элемент управления SplitContainer.

### **Тема 3. Управляющие элементы, обеспечивающие выбор из фиксированного набора альтернатив, предоставленных пользователю.**

Элементы управления – список и комбинированный список. Редактирование списков. Сортировка в списках на этапе проектирования и во время выполнения приложения. Операции поиска в списках и комбинированных списках. Обработка событий, связанных с выбором пользователем строки списка. Обработка множественного выбора в списке. Обработка событий, связанных с вводом пользователем строки в поле комбинированного списка. Свойства списка. Стиль списка. Методы очистки списка, добавления и удаления строк. События, связанные с выбором строк списка и редактированием поля комбинированного списка. Элементы управления – флажки и переключатели (радио-кнопки). Состояние флажков и переключателей. Состояние переключателей в группе. Обработка события - выбора переключателя в группе. Программное изменение состояния переключателя. Программное изменение состояния группы переключателей. Контейнеры – формы, рисунки, рамки. Элементы управления – рисунки, изображения, рамки. Работа с массивами элементов управления

### **Тема 4. Создание и использование многоуровневого меню.**

Клавиши быстрого вызова. Комбинации клавиш – «горячие клавиши» (акселераторы). Управление доступом к пунктам меню. Группировка команд меню. Процедуры обработки команд меню. Создание и применение контекстного всплывающего меню. Активизация контекстного меню. Визуальный компонент – панель инструментов. Создание панелей инструментов. Размещение панели инструментов в окне.

Использование изображений для кнопок панели инструментов. Простые кнопки, кнопки с фиксацией. Группы кнопок: кнопки – переключатели. Кнопки – списки. Программирование функций кнопок панели инструментов.

### **Тема 5. Отображение информации о состоянии программы и визуализация выполняемых операций.**

Последовательность событий при нажатии кнопок мыши и определение модифицирующих клавиш клавиатуры (Alt, Shift ...). Смена образа и положения курсора мыши. Координаты курсора. Технология Drag&Drop. Применение технологии Drag&Drop для обмена информацией между компонентами. Последовательность событий при нажатии клавиш клавиатуры. Определение модифицирующих клавиш клавиатуры. Использование событий клавиатуры для верификации вводимой пользователем информации. Строка состояния. Панели строки состояния. Расположение панелей состояния. Стили панелей. Визуализация длительных процессов.

### **Тема 6. Компоненты, предназначенные для работы с базами данных.**

Подключение к базам данных. Выбор таблицы. Компоненты, предназначенные для отображения значения поля и навигации по базе данных. Отображение информации в табличном виде. Фильтрация и упорядочивание записей базы данных. Компоненты, предназначенные для построения запросов SQL. Подключение к базе данных и настройка. Построение и редактирование запроса по требованиям пользователя приложения. Запросы с параметром.

### **Тема 7. Создание ресурсов приложения.**

Создание файла помощи и его использование в проекте. Подготовка приложения к выпуску. Тестирование и отладка приложений.

### **Тема 8. Зачет.**

Вопросы для подготовки к зачёту 1. Принципы визуального программирования 2. Какие файлы необходимо сохранить для дальнейшей работы с проектом. 3. Какие файлы создаются при компиляции приложения. Требуются ли они при дальнейшей работе с проектом. 4. Инспектор объектов. Свойства и события. 5. Окно проекта. 6. Окно редактора кода. 7. Палитра компонентов. 8. Что такое компонент. 9. Что такое свойства и методы компонента с точки зрения объектно-ориентированного программирования. 10. Что такое форма. 11. В чем отличие между визуальными и не визуальными компонентами. 12. Виды командных кнопок. 13. В чем отличие между метками и текстовыми полями. 14. Какие компоненты предназначены для просмотра и редактирования символьной информации. 15. Какие компоненты предназначены для выбора из фиксированного набора альтернатив, предоставленных пользователю. 16. Чем отличаются возможности, предоставляемые списками и комбинированными списками. 17. Стили списков и комбинированных списков. 18. Какие свойства и методы предназначены для редактирования списков и комбинированных списков. 19. Способы создания группы переключателей. 20. Свойства компонентов, предназначенные для выбора параметров шрифта. 21. Методы редактирования стилей шрифта. 22. События. Обработка событий. Событие по умолчанию. 23. Каким образом можно управлять доступностью, видимостью и положением компонента на форме. 24. Контейнеры. Свойства компонентов, помещенных в контейнер. 25. Чем отличаются группы переключателей и группы флажков. 26. События мыши. Обработка событий, связанных с перемещением и нажатием кнопок мыши. 27. События клавиатуры. Обработка событий, связанных с нажатием клавиш клавиатуры. 28.

Меню. Создание и редактирование вложенного меню. 29. Команды меню. Обработка команд меню. Клавиши быстрого доступа. 30. Контекстное меню. 31. Панели инструментов. Создание, размещение и редактирование панелей инструментов. 32. Использование изображений для кнопок панели инструментов и команд меню. 33. Группы кнопок. Кнопки с фиксацией. Кнопки-списки. 34. Какие компоненты предназначены для навигации по файловой системе компьютера. 35. Встроенные диалоги сообщений. Использование встроенных диалогов для организации ветвления в программе. 36. Стандартные диалоги. 37. Строка состояния. Панели строки состояния. Расположение панелей состояния. Стили панелей. 38. Визуальные компоненты, предназначенные для отображения графических примитивов. 39. Графика. Компоненты, предназначенные для просмотра и редактирования графических изображений. 40. Задание свойств пера и кисти. Рисование линий и примитивов. 41. Режимы рисования и заливки. 42. Страницы свойств и вкладки. 43. Многооконные приложения. Модальные и немодальные диалоги. 44. Компоненты, предназначенные для управления числовыми параметрами. 45. Компоненты, предназначенные для визуализации длительных процессов. 46. Функции, предназначенные для преобразования данных различных типов. 47. Компоненты и функции, предназначенные для работы с датой и временем. 48. Способы верификации вводимой пользователем информации. 49. Какие свойства компонентов предназначены для создания всплывающих подсказок и работы со справочной системой приложения. 50. Работа с буфером обмена. Копирование текстовой и графической информацией. 51. Применение технологии Drag&Drop для обмена информацией между компонентами. 52. Какие компоненты, предназначены для работы с базами данных. 53. Как осуществляется подключение к базам данных, выбор таблиц и полей. 54. Какие свойства компонента позволяют осуществить фильтрацию или упорядочивание данных. 55. Какие компоненты и их свойства позволяют выполнить запрос SQL к подключенной базе данных.

## **Планы практических занятий**

### **заочная форма обучения**

**Тема 3. Управляющие элементы, обеспечивающие выбор из фиксированного набора альтернатив, предоставленных пользователю..** Работа с массивами элементов управления. .

Время - 1 час.

#### Основные вопросы:

Создание массивов. Операции над элементами массивов. Использование встроенных диалоговых окон для организации ветвления в программе.

**Тема 4. Создание и использование многоуровневого меню..** Обеспечение навигации по файлам и папкам..

Время - 1 час.

#### Основные вопросы:

Обеспечение навигации по файлам и папкам. Атрибуты файлов. Флаги – «только чтение. Тип (расширение) по умолчанию. Использование событий клавиатуры для верификации вводимой пользователем информации. Визуализация длительных процессов.

**Тема 6. Компоненты, предназначенные для работы с базами данных.** Подключение к базе данных и настройка форм для работы с базой..

Время - 1 час.

#### Основные вопросы:

Отображение информации в табличном виде. Построение и редактирование запроса по требованиям пользователя приложения. Запросы с параметром.

Тема 7. **Создание ресурсов приложения..** Создание справки на основе файлов в формате RTF( в формате HTML).

Время - 1 час.

Основные вопросы:

Создание содержания справочной системы.

## **6. ПЕРЕЧЕНЬ УЧЕБНО-МЕТОДИЧЕСКОГО ОБЕСПЕЧЕНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ**

6.1.1. Основные категории учебной дисциплины для самостоятельного изучения:

**Язык программирования** - формальный язык, предназначенный для записи компьютерных программ. Язык программирования определяет набор лексических, синтаксических и семантических правил, определяющих внешний вид программы и действия, которые выполнит исполнитель (обычно - ЭВМ) под её управлением.

**Алфавит языка** - основные символы языка - буквы, цифры и специальные символы.

**Лексема** - последовательность допустимых символов языка программирования, имеющая смысл для транслятора.

**Идентификатор (ID)** – это имя программного объекта (константы, переменной, метки, типа, функции и т.д.). В идентификаторе могут использоваться латинские буквы, цифры и знак подчеркивания; первый символ ID – не цифра; пробелы внутри ID не допускаются.

**Ключевые слова** - идентификаторы, смысл которого зафиксирован правилами языка программирования и который используется для распознавания предложений в программе

**Псевдокод** - компактный (зачастую неформальный) язык описания алгоритмов, использующий ключевые слова императивных языков программирования, но опускающий несущественные подробности и специфический синтаксис. Псевдокод обычно опускает детали, несущественные для понимания алгоритма человеком. Такими несущественными деталями могут быть описания переменных, системно-зависимый код и подпрограммы.

**Блок-схема** - распространенный тип схем (графических моделей), описывающих алгоритмы или процессы, в которых отдельные шаги изображаются в виде блоков различной формы, соединенных между собой линиями, указывающими направление последовательности.

**Трансляция программы** - преобразование программы, представленной на одном из языков программирования, в машинный код или, иногда, в язык ассемблера. Транслятор обычно выполняет также диагностику ошибок, формирует словари идентификаторов, выдаёт для печати текст программы и т. д.

**Транслятор** - программа или техническое средство, выполняющее трансляцию программы.

**Компоновщик** (также редактор связей, линкер - от англ. link editor, linker) - программа, которая производит компоновку - принимает на вход один или несколько объектных модулей и собирает по ним исполнимый модуль. Для связывания модулей, компоновщик использует таблицы имён, созданные компилятором в каждом из объектных модулей.

**Компилировать** - проводить трансляцию машинной программы с проблемно-ориентированного языка на машинно-ориентированный язык

**Сборка** (англ. build) (имя существительное) - подготовленный для использования информационный продукт. Чаще всего сборка - исполняемый файл - двоичный файл, содержащий исполняемый код (машинные инструкции) программы или библиотеки.



**Сборка** (англ. build) (глагол) - процесс получения информационного продукта из исходного кода. Чаще всего включает компиляцию и компоновку, выполняется инструментами автоматизации.

**Листинг** - исходный код (также исходный текст) – текст компьютерной программы на каком-либо языке программирования. В обобщённом смысле любые входные данные для транслятора. Исходный код либо транслируется в исполняемый код при помощи компилятора, либо исполняется непосредственно по тексту при помощи интерпретатора.

**Исполняемый код** (Машинный код, платформенно-ориентированный код), машинный язык — система команд (набор кодов операций) конкретной вычислительной машины, которая интерпретируется непосредственно процессором или микропрограммами этой вычислительной машины. Компьютерная программа, записанная на машинном языке, состоит из машинных инструкций, каждая из которых представлена в машинном коде в виде т. н. опкода — двоичного кода отдельной операции из системы команд машины. Для удобства программирования вместо числовых опкодов, которые только и понимает процессор, обычно используют их условные буквенные мнемоники. Набор таких мнемоник, вместе с некоторыми дополнительными возможностями (например, некоторыми макрокомандами, директивами), называется языком ассемблера.

**Инструкция или оператор** (англ. statement) - наименьшая автономная часть языка программирования; команда или набор команд. Программа обычно представляет собой последовательность инструкций. Многие языки (например, Си) различают инструкцию и определение. Различие в том, что инструкция исполняет код, а определение создаёт идентификатор.

**Точка входа** (англ. Entry Point (EP) - точка входа) - адрес в оперативной памяти, с которого начинается выполнение программы. Другими словами - адрес, по которому хранится первая команда программы. Однако не надо путать её с «первыми командами» программы на языке высокого уровня. Например, программа на C++ начинает выполнение с функции main(), на самом деле, программа в памяти начинается с первой исполняемой инструкции этой функции.

**Интегрированная среда разработки, ИСР** (англ. Integrated development environment - IDE), также единая среда разработки, ЕСР - комплекс программных средств, используемый программистами для разработки программного обеспечения (ПО). Среда разработки включает в себя: текстовый редактор, компилятор и/или интерпретатор, средства автоматизации сборки, отладчик.

**Отладка** - этап разработки компьютерной программы, на котором обнаруживают, локализируют и устраняют ошибки. Чтобы понять, где возникла ошибка, приходится:

**Отладчик** - это программный модуль, который позволяет выполнить основные задачи, связанные с мониторингом процесса выполнения результирующей прикладной программы. Этот процесс называется отладкой и включает в себя следующие основные возможности: последовательное пошаговое выполнение результирующей программы на основе шагов по машинным командам или по операторам входного языка; выполнение результирующей программы до достижения ею одной из заданных точек останова (адресов останова); выполнение результирующей программы до наступления некоторых заданных условий, связанных с данными и адресами, обрабатываемыми этой программой; просмотр содержимого областей памяти, занятых командами или данными результирующей программы.

**Трассировка** - процесс пошагового выполнения программы. В режиме трассировки программист видит последовательность выполнения команд и значения переменных на данном шаге выполнения программы, что позволяет легче обнаруживать ошибки. Трассировка может быть начата и окончена в любом месте программы, выполнение программы может останавливаться на каждой команде или на точках останова, трассировка может выполняться с заходом в процедуры и без заходов, а также осуществляться в обратном порядке (шаг назад).

**Визуальное программирование** - способ создания программы для ЭВМ путём манипулирования графическими объектами вместо написания её текста. Визуальное программирование часто представляют как следующий этап развития текстовых языков программирования.

**Визуальные средства разработки** - как правило, под ними подразумевают средства проектирования интерфейсов или какую либо CASE-систему для быстрой разработки приложений.

**Компонент** - в программировании, множество классов и языковых конструкций, объединенных по общему признаку. В большинстве языков программирования нет языковых конструкций прямо отражающих понятие компонента. Компоненты реализуются с помощью стандартных конструкций, таких как классы

**UML** – унифицированный язык моделирования (Unified Modeling Language) – это система обозначений, которую можно применять для объектно-ориентированного анализа и проектирования. Его можно использовать для визуализации, спецификации, конструирования и документирования программных систем.

**STL** – (Standard Template Library) является программной библиотекой для C++ языка программирования, который влиял на многие части стандартной библиотеки C++. Он предоставляет четыре компонента, которые называются алгоритмами, контейнерами, функциями и итераторами.

**Библиотека визуальных компонентов** (англ. Visual Component Library, VCL) - объектно-ориентированная библиотека для разработки программного обеспечения, разработанная компанией Borland (на данный момент поддерживается Embarcadero) для поддержки принципов визуального программирования. VCL входит в комплект поставки Delphi, C++ Builder и Embarcadero RAD Studio и является, по сути, частью среды разработки, хотя разработка приложений в этих средах возможна и без использования VCL. VCL предоставляет огромное количество готовых к использованию компонентов для работы в самых разных областях программирования, таких, например, как интерфейс пользователя (экранные формы и элементы управления - т. н. «контролы», «контроли»), работа с базами данных, взаимодействие с операционной системой, программирование сетевых приложений и прочее. Net.

**Net** - .NET Framework — программная платформа, выпущенная компанией Microsoft в 2002 году. Основой платформы является общезыковая среда исполнения Common Language Runtime (CLR), которая подходит для разных языков программирования. Функциональные возможности CLR доступны в любых языках программирования, использующих эту среду.

**CLR** - Common Language Runtime (англ. CLR - общезыковая исполняющая среда) - исполняющая среда для байт-кода CIL (MSIL), в которой компилируются программы, написанные на .NET-совместимых языках программирования (C#, Managed C++, Visual Basic .NET, F# и прочие). CLR является одним из основных компонентов пакета Microsoft .NET Framework

**CLS** - Common Language Specification (сокр. CLS, рус. Общезыковая спецификация) - документ, в котором говорится, как компьютерные программы могут быть превращены в код MSIL

**CTS** - Common Type System (сокр. CTS, рус. Общая система типов) - часть .NET Framework, формальная спецификация, определяющая, как какой-либо тип (класс, интерфейс, структура, встроенный тип данных) должен быть определён для его правильного выполнения средой .NET. Кроме того, данный стандарт определяет, как определения типов и специальные значения типов представлены в компьютерной памяти. Целью разработки CTS было обеспечение возможности программам, написанным на различных языках программирования, легко обмениваться информацией. Как это принято в языках программирования, тип может быть описан как определение набора допустимых

значений (например, «все целые от 0 до 10») и допустимых операций над этими значениями (например, сложение и вычитание).

**Сборка .Net** - имеют следующие составляющие: Манифест, который содержит метаданные сборки. Метаданные типов. Используя эти метаданные, сборка определяет местоположение типов в файле приложения, а также места размещения их в памяти. Собственно код приложения на языке MSIL, в который компилируется код C#. Ресурсы. Все эти компоненты могут находиться в одном файле, и тогда сборка представляет один единственный файл в формате exe или dll.

**COM** (англ. Component Object Model - модель компонентного объекта; произносится как [ком]) - это технологический стандарт от компании Microsoft, предназначенный для создания программного обеспечения на основе взаимодействующих компонентов объекта, каждый из которых может использоваться во многих программах одновременно. Стандарт воплощает в себе идеи полиморфизма и инкапсуляции объектно-ориентированного программирования. Стандарт COM мог бы быть универсальным и платформо-независимым, но закрепился в основном на операционных системах семейства Microsoft Windows. В современных версиях Windows COM используется очень широко. На основе COM были реализованы технологии: Microsoft OLE Automation, ActiveX, DCOM, COM+, DirectX, а также XPCOM.

**CIL (MSIL, IL)** - Common Intermediate Language (сокращённо CIL) - «высокоуровневый ассемблер» виртуальной машины .NET. Промежуточный язык, разработанный фирмой Microsoft для платформы .NET Framework. JIT-компилятор CIL является частью CLR (англ. common language runtime) - общей среды выполнения программ, написанных на языках .NET. Ранее язык назывался «Microsoft Intermediate Language (MSIL)», однако был переименован для создания стандарта «ECMA-335».

Все компиляторы, поддерживающие платформу .NET, должны транслировать код с языков высокого уровня платформы .NET на язык CIL. В частности, код на языке CIL генерируют все компиляторы .NET фирмы Microsoft, входящие в среду разработки Microsoft Visual Studio (C#, Managed C++, Visual Basic .NET, Visual J# .NET).

**ADO** (от англ. ActiveX Data Objects - «объекты данных ActiveX») - интерфейс программирования приложений для доступа к данным, разработанный компанией Microsoft (MS Access, MS SQL Server) и основанный на технологии компонентов ActiveX. ADO позволяет представлять данные из разнообразных источников (реляционных баз данных, текстовых файлов и т. д.) в объектно-ориентированном виде. ADO.NET (ActiveX Data Object для .NET) - технология, предоставляющая доступ и управление данными, хранящимся в базе данных или других источниках (Microsoft SQL Server, Microsoft Access, Microsoft Excel, Microsoft Outlook, Microsoft Exchange, Oracle, OLE DB, ODBC, XML, текстовые файлы)[1], основанных на платформе .NET Framework и входящая в состав .NET Framework 2.0, представляет собой набор библиотек[2]. В отличие от технологии ADO, которая была в основном предназначена для тесно связанных клиент-серверных систем, ADO.NET больше нацелена на автономную работу с помощью объектов DataSet. Объекты DataSet представляют локальные копии взаимосвязанных таблиц данных, каждая из которых содержит набор строк и столбцов. Объекты DataSet позволяют вызывающей сборке (наподобие веб-страницы или программы, выполняющейся на настольном компьютере) работать с содержимым DataSet, изменять его, не требуя подключения к источнику данных, и отправлять обратно блоки измененных данных для обработки с помощью соответствующего адаптера данных. Но, пожалуй, самое фундаментальное различие между классической ADO и ADO.NET состоит в том, что ADO.NET является управляемой кодовой библиотекой, и, значит, подчиняется тем же правилам, что и любая управляемая библиотека. Типы, составляющие ADO.NET, используют протокол управления памятью CLR, принадлежат к той же системе типов (классы, интерфейсы, перечисления, структуры и делегаты), и доступ к ним возможен с помощью любого языка .NET. Классы ADO.NET находятся в сборке System.Data.dll.

**XML** (/ˌɛks em ˈel/ англ. eXtensible Markup Language) - расширяемый язык разметки. Рекомендован Консорциумом Всемирной паутины (W3C). Спецификация XML описывает XML-документы и частично описывает поведение XML-процессоров (программ, читающих XML-документы и обеспечивающих доступ к их содержимому). XML разрабатывался как язык с простым формальным синтаксисом, удобный для создания и обработки документов программами и одновременно удобный для чтения и создания документов человеком, с подчёркиванием нацеленности на использование в Интернете. Язык называется расширяемым, поскольку он не фиксирует разметку, используемую в документах: разработчик волен создать разметку в соответствии с потребностями к конкретной области, будучи ограниченным лишь синтаксическими правилами языка. Расширение XML — это конкретная грамматика, созданная на базе XML и представленная словарём тегов и их атрибутов, а также набором правил, определяющих какие атрибуты и элементы могут входить в состав других элементов. Сочетание простого формального синтаксиса, удобства для человека, расширяемости, а также базирование на кодировках Юникод для представления содержания документов привело к широкому использованию как собственно XML, так и множества производных специализированных языков на базе XML в самых разнообразных программных средствах.

**XAML** (англ. eXtensible Application Markup Language) - расширяемый язык разметки для приложений (произносится [замл] или [зэмл]) - основанный на XML язык разметки для декларативного программирования приложений, разработанный Microsoft.

Модель приложений Vista включает объект Application. Его набор свойств, методов и событий позволяет объединить веб-документы в связанное приложение. Объект Application контролирует выполнение программы и генерирует события для пользовательского кода. Документы приложения пишутся на XAML. С помощью XAML описывается, прежде всего, пользовательский интерфейс. Логика приложения по-прежнему управляется процедурным кодом (C#, VB, JavaScript и т. д.). XAML может использоваться как для браузер-базированных приложений, так и для настольных приложений.

#### 6.1.2 Задания для приобретения, углубления и закрепления приобретаемых знаний

№	Задание	Код результата обучения
1.	1.Принципы визуального программирования. 2.Приведите понятие и примеры визуальных объектов 3.Перечислите общие методы и свойства форм и элементов управления	ПК-3–31
2.	4.Работа с буфером обмена. Копирование текстовой и графической информации. 5.Применение технологии Drag&Drop для обмена информацией между компонентами.	ПК-3–31
3.	6.Что такое компонент? Что такое палитра компонентов? 7.Какие компоненты предназначены для выбора из фиксированного набора альтернатив, предоставленных пользователю. 8.Поясните свойства и методы компонента с точки зрения объектно-ориентированного программирования.	ПК-3–32
4.	9.Какие компоненты предназначены для выбора из фиксированного набора альтернатив, предоставленных пользователю. 10.Поясните, как выполнить подключение к базе данных и настройку форм приложения для работы с базой.	ПК-3–32
5.	11.Поясните отображение информации в табличном виде. 12.Что такое запрос к базе данных? Запросы с параметром.	ПК-3–33

6.	13.Поясните методы проектирования ИС различного профиля	ПК-3–33
7.	14.Как провести изменение внешнего вида и поведения формы приложения? 15.Перечислите и поясните свойства компонентов, помещенных в контейнер.	ПК-3–34
8.	16.Поясните создание и редактирование меню, вложенного меню. 17.Перечислите стандартные команды меню. 18.Как выполнить обработку команд меню? Клавиши быстрого доступа.	ПК-3–34

6.2. Задания, направленные на формирование профессиональных умений.

№	Задание	Код результата обучения
9.	1. Выполните разметку окна приложения в соответствии с заданием преподавателя с помощью дизайнера окон ( Form Designer ) или путем изменения свойств в окне Свойства ( Properties ) или в коде программы. Выполните добавление/удаление формы в приложение во время выполнения.	ПК-3–У1
10.	2. Измените внешний вид и поведение формы. Добавьте в конструкторе элемент управления в форму или контейнерный элемент управления.	ПК-3–У1
11.	3.Добавьте в форму две кнопки (1 и 2), для которых задать различные цвета (свойство BackColor).	ПК-3–У2
12.	4. Напишите для кнопок 1 и 2 обработчики, которые изменяют цвета кнопок: при неоднократном нажатии любой кнопки цвета кнопок меняются ( цвет кнопки 1 меняется на цвет кнопки 2 и наоборот). Добавьте кнопку "Выход", используя метод Exit() класса Application.	ПК-3–У2
13.	5. Создайте главное меню, включающее следующие пункты: "Объект", "Справочник", "Справка". Для пункта "Объект" создайте следующие подпункты: "Сотрудник", "Клиент", "Договор", "Поручение", "Сделка", "Выход".	ПК-3–У3
14.	6. Для пункта меню "Справочник" создайте следующие подпункты: "Должность", "Страна", "Регион", "Город", "ООО Тур". Для пункта "Справка" создать подпункт - "О программе".	ПК-3–У3
15.	7. Разработайте методы для задания режимов "Просмотр", "Редактирование" для элементов контроля формы приложения.	ПК-3–У4
16.	8. Разработайте методы для задания режимов "Просмотр", "Редактирование" для управления активностью пунктов главного меню формы, контекстного меню и кнопок панели инструментов.	ПК-3–У4

6.3. Задания, направленные на формирование профессиональных навыков, владений.

№	Задание	Код результата обучения
17.	1.Создайте в приложении клиента объект DataSet для работы с таблицами базы данных.	ПК-3–В1
18.	2. Для разрабатываемого приложения объекты dsEmployee, daJobTitle и daEmployee. Разработайте метод Fill для заполнения таблиц DataSet.	ПК-3–В1
19.	3. Разработайте методы для модификации, формирования, удаления и сохранения данных по таблице Сотрудники объекта DataSet.	ПК-3–В2
20.	4. Создайте компоненты приложения, для отображения значений и	ПК-3–В2

	навигации по базе данных. Выполните отображение информации в табличном виде.	
21.	5. Разработайте метод для отмены модификации данных по сотрудникам объекта DataSet, используя метод метода RejectChanges.	ПК-3–В3
22.	6. Выполните упорядочивание списка сотрудников по алфавиту, обрабатывая данные получаемые из базы данных с помощью SQL - запроса(команда Select).	ПК-3–В3
23.	7.Модифицируйте приложение для вывода списка сотрудников, содержащего фамилии, имени и отчества в визуальные компоненты приложения.	ПК-3–В4
24.	8.Создайте файл помощи и подключите его в приложение	ПК-3–В4

## 7. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ ТЕКУЩЕГО КОНТРОЛЯ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

### 7.1. Средства оценивания в ходе текущего контроля:

- письменные краткие опросы в ходе аудиторных занятий на знание категорий учебной дисциплины, указанных в п.6.1.;
- задания и упражнения, рекомендованные для самостоятельной работы;
- практические работы;
- ответы на вопросы к зачету.

### 7.2. ФОС для текущего контроля

№	Код результата обучения	ФОС текущего контроля
1.	ПК-3-31	Задания для самостоятельной работы 1-3
2.	ПК-3-31	Задания для самостоятельной работы 4-5
3.	ПК-3-32	Задания для самостоятельной работы 6-8
4.	ПК-3-32	Задания для самостоятельной работы 9-10
5.	ПК-3-33	Задания для самостоятельной работы 11-12
6.	ПК-3-33	Задания для самостоятельной работы 13
7.	ПК-3-34	Задания для самостоятельной работы 14-16
8.	ПК-3-34	Задания для самостоятельной работы 17-18
9.	ПК-3-У1	Задание 1 п. 6.2 Задания, направленные на формирование профессиональных умений
10.	ПК-3-У1	Задание 2 п. 6.2 Задания, направленные на формирование профессиональных умений
11.	ПК-3-У2	Задание 3 п. 6.2 Задания, направленные на формирование профессиональных умений
12.	ПК-3-У2	Задание 4 п. 6.2 Задания, направленные на формирование профессиональных умений
13.	ПК-3-У3	Задание 5 п. 6.2 Задания, направленные на формирование профессиональных умений
14.	ПК-3-У3	Задание 6 п. 6.2 Задания, направленные на формирование профессиональных умений
15.	ПК-3-У4	Задание 7 п. 6.2 Задания, направленные на формирование профессиональных умений
16.	ПК-3-У4	Задание 8 п. 6.2 Задания, направленные на

		формирование профессиональных умений
17.	ПК-3-B1	Задание 1 п. 6.3 Задания направленные на формирования профессиональных навыков
18.	ПК-3-B1	Задание 2 п. 6.3 Задания направленные на формирования профессиональных навыков
19.	ПК-3-B2	Задание 3 п. 6.3 Задания направленные на формирования профессиональных навыков
20.	ПК-3-B2	Задание 4 п. 6.3 Задания направленные на формирования профессиональных навыков
21.	ПК-3-B3	Задание 5 п. 6.3 Задания направленные на формирования профессиональных навыков
22.	ПК-3-B3	Задание 6 п. 6.3 Задания направленные на формирования профессиональных навыков
23.	ПК-3-B4	Задание 7 п. 6.3 Задания направленные на формирования профессиональных навыков
24.	ПК-3-B4	Задание 8 п. 6.3 Задания направленные на формирования профессиональных навыков

### 7.3. ФОС для промежуточной аттестации

#### Задания для оценки знаний

№	Код результата обучения	Задания
1.	ПК-3-31	1. Принципы визуального программирования 2. Какие файлы необходимо сохранить для дальнейшей работы с проектом. 3. Какие файлы создаются при компиляции приложения. Требуются ли они при дальнейшей работе с проектом. 4. Инспектор объектов. Свойства и события. 5. Окно проекта. 6. Окно редактора кода.
2.	ПК-3-31	7. Палитра компонентов. 8. Что такое компонент. 9. Что такое свойства и методы компонента с точки зрения объектно-ориентированного программирования. 10. Что такое форма.
3.	ПК-3-32	11. В чем отличие между визуальными и не визуальными компонентами. 12. Виды командных кнопок. 13. В чем отличие между метками и текстовыми полями. 14. Какие компоненты предназначены для просмотра и редактирования символьной информации. 15. Какие компоненты предназначены для выбора из фиксированного набора альтернатив, предоставленных пользователю.
4.	ПК-3-32	16. Чем отличаются возможности, предоставляемые списками и комбинированными списками. 17. Стили списков и комбинированных списков. 18. Какие свойства и методы предназначены для редактирования списков и комбинированных списков. 19. Способы создания группы переключателей. 20. Свойства компонентов, предназначенные для выбора параметров шрифта.
5.	ПК-3-33	22. События. Обработка событий. Событие по умолчанию. 23. Каким образом можно управлять доступностью, видимостью и положением

		компонента на форме. 24. Контейнеры. Свойства компонентов, помещенных в контейнер. 25. Чем отличаются группы переключателей и группы флажков. 26. События мыши. Обработка событий, связанных с перемещением и нажатием кнопок мыши. 27. События клавиатуры. Обработка событий, связанных с нажатием клавиш клавиатуры. 28. Меню. Создание и редактирование вложенного меню. 29. Команды меню. Обработка команд меню. Клавиши быстрого доступа. 30. Контекстное меню.
6.	ПК-3-33	31. Панели инструментов. Создание, размещение и редактирование панелей инструментов. 32. Использование изображений для кнопок панели инструментов и команд меню. 33. Группы кнопок. Кнопки с фиксацией. Кнопки-списки. 34. Какие компоненты предназначены для навигации по файловой системе компьютера. 35. Встроенные диалоги сообщений. Использование встроенных диалогов для организации ветвления в программе. 36. Стандартные диалоги. 37. Строка состояния. Панели строки состояния. Расположение панелей состояния. Стили панелей. 38. Визуальные компоненты, предназначенные для отображения графических примитивов. 39. Графика. Компоненты, предназначенные для просмотра и редактирования графических изображений. 40. Задание свойств пера и кисти. Рисование линий и примитивов. 41. Режимы рисования и заливки.
7.	ПК-3-34	42. Страницы свойств и вкладки. 43. Многооконные приложения. Модальные и немодальные диалоги. 44. Компоненты, предназначенные для управления числовыми параметрами. 45. Компоненты, предназначенные для визуализации длительных процессов. 46. Функции, предназначенные для преобразования данных различных типов. 47. Компоненты и функции, предназначенные для работы с датой и временем. 48. Способы верификации вводимой пользователем информации. 49. Какие свойства компонентов предназначены для создания всплывающих подсказок и работы со справочной системой приложения.
8.	ПК-3-34	50. Работа с буфером обмена. Копирование текстовой и графической информацией. 51. Применение технологии Drag&Drop для обмена информацией между компонентами. 52. Какие компоненты, предназначены для работы с базами данных. 53. Как осуществляется подключение к базам данных, выбор таблиц и полей. 54. Какие свойства компонента позволяют осуществить фильтрацию или упорядочивание данных. 55. Какие компоненты и их свойства позволяют выполнить запрос SQL к подключенной базе данных.

### Задания для оценки умений



№	Код результата обучения	Задания
1.	ПК-3-У1	В качестве фондов оценочных средств для оценки умений обучающегося используются задания 1, рекомендованные для выполнения в часы самостоятельной работы (раздел 6.2.)
2.	ПК-3-У1	В качестве фондов оценочных средств для оценки умений обучающегося используются задания 2, рекомендованные для выполнения в часы самостоятельной работы (раздел 6.2.).
3.	ПК-3-У2	В качестве фондов оценочных средств для оценки умений обучающегося используются задания 3, рекомендованные для выполнения в часы самостоятельной работы (раздел 6.2.).
4.	ПК-3-У2	В качестве фондов оценочных средств для оценки умений обучающегося используются задания 4, рекомендованные для выполнения в часы самостоятельной работы (раздел 6.2.).
5.	ПК-3-У3	В качестве фондов оценочных средств для оценки умений обучающегося используются задания 4, рекомендованные для выполнения в часы самостоятельной работы (раздел 6.2.).
6.	ПК-3-У3	В качестве фондов оценочных средств для оценки умений обучающегося используются задания 5, рекомендованные для выполнения в часы самостоятельной работы (раздел 6.2.).
7.	ПК-3-У4	В качестве фондов оценочных средств для оценки умений обучающегося используются задания 6, рекомендованные для выполнения в часы самостоятельной работы (раздел 6.2.).
8.	ПК-3-У4	В качестве фондов оценочных средств для оценки умений обучающегося используются задания 7, рекомендованные для выполнения в часы самостоятельной работы (раздел 6.2.).

**Задания, направленные на формирование профессиональных навыков, владений.**

№	Код результата обучения	Задания
1.	ПК-3-В1	В качестве фондов оценочных средств для оценки навыков, владений, опыта, деятельности обучающегося используются задания 1, рекомендованные для выполнения в часы самостоятельной работы (раздел 6.3), а также практические работы
2.	ПК-3-В1	В качестве фондов оценочных средств для оценки навыков, владений, опыта, деятельности обучающегося используются задания 2, рекомендованные для выполнения в часы самостоятельной работы (раздел 6.3), а также практические работы
3.	ПК-3-В2	В качестве фондов оценочных средств для оценки навыков, владений, опыта, деятельности обучающегося используются задания 3,

		рекомендованные для выполнения в часы самостоятельной работы (раздел 6.3), а также практические работы
4.	ПК-3-В2	В качестве фондов оценочных средств для оценки навыков, владений, опыта, деятельности обучающегося используются задания 4, рекомендованные для выполнения в часы самостоятельной работы (раздел 6.3), а также практические работы
5.	ПК-3-В3	В качестве фондов оценочных средств для оценки навыков, владений, опыта, деятельности обучающегося используются задания 5, рекомендованные для выполнения в часы самостоятельной работы (раздел 6.3), а также практические работы
6.	ПК-3-В3	В качестве фондов оценочных средств для оценки навыков, владений, опыта, деятельности обучающегося используются задания 6, рекомендованные для выполнения в часы самостоятельной работы (раздел 6.3), а также практические работы
7.	ПК-3-В4	В качестве фондов оценочных средств для оценки навыков, владений, опыта, деятельности обучающегося используются задания 7, рекомендованные для выполнения в часы самостоятельной работы (раздел 6.3), а также практические работы
8.	ПК-3-В4	В качестве фондов оценочных средств для оценки навыков, владений, опыта, деятельности обучающегося используются задания 8, рекомендованные для выполнения в часы самостоятельной работы (раздел 6.3), а также практические работы

## 8. ПЕРЕЧЕНЬ ОСНОВНОЙ И ДОПОЛНИТЕЛЬНОЙ УЧЕБНОЙ ЛИТЕРАТУРЫ, НЕОБХОДИМОЙ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ).

### Основная литература:

1. Жмудь, В. А. Моделирование замкнутых систем автоматического управления : учебное пособие для вузов / В. А. Жмудь. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2024. — 128 с. — (Высшее образование). — ISBN 978-5-534-09487-9. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/538826>
2. Огнева, М. В. Программирование на языке C++: практический курс : учебное пособие для вузов / М. В. Огнева, Е. В. Кудрина. — Москва : Издательство Юрайт, 2024. — 335 с. — (Высшее образование). — ISBN 978-5-534-05123-0. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/539713>

### Дополнительная литература:

1. Черткова, Е. А. Программная инженерия. Визуальное моделирование программных систем : учебник для вузов / Е. А. Черткова. — 3-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2024. — 146 с. — (Высшее образование). — ISBN 978-5-534-

- 18197-5. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/534516>
2. Мерзлякова, Е. Ю. Визуальное программирование и человеко-машинное взаимодействие / Е. Ю. Мерзлякова. — Новосибирск : Сибирский государственный университет телекоммуникаций и информатики, 2022. — 49 с. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/125262.html>
  3. Козырева Г.Ф. Функциональное и логическое программирование [Электронный ресурс] : учебно-методическое пособие / Г.Ф. Козырева. — Электрон. текстовые данные. — Саратов: Ай Пи Эр Медиа, 2018. — 120 с. — 978-5-4486-0122-4. — Режим доступа: <http://www.iprbookshop.ru/71596.html>

## **9. ПЕРЕЧЕНЬ КОМПЛЕКТОВ ЛИЦЕНЗИОННОГО И СВОБОДНО РАСПРОСТРАНЯЕМОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, ИСПОЛЬЗУЕМОГО ПРИ ИЗУЧЕНИИ УЧЕБНОЙ ДИСЦИПЛИНЫ**

При изучении учебной дисциплины (в том числе в интерактивной форме) предполагается применение современных информационных технологий. Комплект программного обеспечения для их использования включает в себя: операционная система Microsoft Windows 7 Pro, офисный пакет программ Microsoft Office Professional Plus 2010, офисный пакет программ Microsoft Office Professional Plus 2007, антивирусная программа Dr. Web Desktop Security Suite, архиватор 7-zip, аудиопроигрыватель AIMP, просмотр изображений FastStone Image Viewer, ПО для чтения файлов формата PDF Adobe Acrobat Reader, ПО для сканирования документов NAPS2, ПО для записи видео и проведения видеотрансляций OBS Studio, ПО для удалённого администрирования Aspiа, правовой справочник Гарант Аэро, онлайн-версия КонсультантПлюс: Студент, электронно-библиотечная система IPRBooks, электронно-библиотечная система Юрайт, математические вычисления Mathcad 14 University, версия 1С для использования типовых конфигураций в учебных целях: 1С: Предприятие 8. Комплект для обучения в высших и средних учебных заведениях, моделирование бизнес-процессов CA ERwin Process Modeler 7.3, версия 1С для обучения программированию: 1С: Предприятие 8.2 Версия для обучения программированию

## **10. ПЕРЕЧЕНЬ РЕСУРСОВ ИНФОРМАЦИОННО-ТЕЛЕКОММУНИКАЦИОННОЙ СЕТИ «ИНТЕРНЕТ», НЕОБХОДИМЫХ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ).**

### **10.1. Интернет- ресурсы**

1. <http://alistair.cockburn.us> / - Персональная страница Алистера Кокберна – специалиста по организации и методологии управления проектами по разработке ПО.
2. <http://www.martinfowler.com> - Персональная страница Мартина Фаулера – известного специалиста в области технологии и методологии разработки ПО.
3. ЭБС IPRbooks (АйПиАрбукс) <http://www.iprbookshop.ru>
4. Образовательная платформа ЮРАЙТ <https://urait.ru>

## **11. ОСОБЕННОСТИ ОБУЧЕНИЯ ИНВАЛИДОВ И ЛИЦ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ.**

Изучение учебной дисциплины «Визуальное программирование» обучающимися инвалидами и лицами с ограниченными возможностями здоровья осуществляется в соответствии с Приказом Министерства образования и науки РФ от 9 ноября 2015 г. № 1309 «Об утверждении Порядка обеспечения условий доступности для инвалидов

объектов и предоставляемых услуг в сфере образования, а также оказания им при этом необходимой помощи» (с изменениями и дополнениями), Методическими рекомендациями по организации образовательного процесса для обучения инвалидов и лиц с ограниченными возможностями здоровья в образовательных организациях высшего образования, в том числе оснащенности образовательного процесса, утвержденными Министерством образования и науки РФ 08.04.2014г. № АК-44/05вн, Положением об организации обучения студентов – инвалидов и лиц с ограниченными возможностями здоровья, утвержденным приказом ректора Университета от 6 ноября 2015 года №60/о, Положением о Центре инклюзивного образования и психологической помощи АНО ВО «Российский новый университет», утвержденного приказом ректора от 20 мая 2016 года № 187/о.

Лица с ограниченными возможностями здоровья и инвалиды обеспечиваются электронными образовательными ресурсами, адаптированными к состоянию их здоровья.

Предоставление специальных технических средств обучения коллективного и индивидуального пользования, подбор и разработка учебных материалов для обучающихся с ограниченными возможностями здоровья производится преподавателями с учетом индивидуальных психофизиологических особенностей обучающихся и специфики приема-передачи учебной информации на основании просьбы, выраженной в письменной форме.

С обучающимися по индивидуальному плану или индивидуальному графику проводятся индивидуальные занятия и консультации.

## **12. ОПИСАНИЕ МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЙ БАЗЫ, НЕОБХОДИМОЙ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ (МОДУЛЮ)**

Учебная аудитория для проведения занятий лекционного типа, занятий семинарского типа, курсового проектирования (выполнения курсовых работ), групповых и индивидуальных консультаций, текущего контроля успеваемости и промежуточной аттестации

Ауд.305 (компьютерный класс №3)

Специализированная мебель:

- столы студенческие;
- стулья студенческие;
- стол для преподавателя;
- стул для преподавателя;
- столы компьютерные;
- кресла компьютерные;
- шкаф для хранения раздаточного материала;
- доска (меловая);
- маркерная доска (переносная).

Технические средства обучения:

- проектор (портативный);
- ПК для преподавателя с возможностью подключения к сети "Интернет" и обеспечением доступа в электронную информационно-образовательную среду вуза;
- ПК для обучающихся с возможностью подключения к сети "Интернет" и обеспечением доступа в электронную информационно-образовательную среду;
- веб-камера;
- экран (переносной);
- колонки;
- микрофон.

Специализированное оборудование:

- наглядные пособия (плакаты), информационный стенд

Автор (составитель): С.В. Толоконников



(подпись)