

**Министерство сельского хозяйства Российской Федерации  
Федеральное государственное, бюджетное образовательное  
учреждение высшего образования «Дагестанский государственный  
аграрный университет имени М.М. Джамбулатова»  
Аграрно-экономический техникум**



**РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ**

**ПМ.05 ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ИНФОРМАЦИОННЫХ СИСТЕМ  
МДК.05.02 «Разработка кода информационных систем»**

**для специальности:**

**09.02.07 «Информационные системы и программирование»**

**Форма обучения – очная**

*Срок обучения СПО по ППССЗ – 2 г.10 м.*

**Махачкала 2023**

Рабочая программа дисциплины разработана на основе Федерального государственного образовательного стандарта по специальности (профессии) среднего профессионального образования для специальности **09.02.07**

**«Информационные системы и программирование», утвержденного приказом Министерства образования и науки Российской Федерации от 9 декабря 2016 г. № 1547.**

Организация-разработчик: ФГБОУ ВО «Дагестанский государственный аграрный университет имени М.М. Джамбулатова» Аграрно-экономический техникум.

**СОГЛАСОВАНО:**



Директор АЭТ

подпись

Магомедов Д.А.

Одобрено на заседании ПЦК  
Общепрофессиональных и  
специальных дисциплин по  
специальности 09.02.07  
«Информационные системы и  
программирование »  
«10» марта 2023г., протокол № 7



Председатель ПЦК

О.О. Касимовская

**СОГЛАСОВАНО:**

Директор Компании Color- IT, Интернет решения



Салихов А.Б.

Ф.И.О.

## **СОДЕРЖАНИЕ**

<b>1. ОБЩАЯ ХАРАКТЕРИСТИКА РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ «МДК.05.02 РАЗРАБОТКА КОДА ИНФОРМАЦИОННЫХ СИСТЕМ» .....</b>	<b>4</b>
<b>2. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ .....</b>	<b>5</b>
<b>3. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ .....</b>	<b>9</b>
<b>4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ .....</b>	<b>9</b>

# 1. ОБЩАЯ ХАРАКТЕРИСТИКА ПРИМЕРНОЙ РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ «МДК.05.02 РАЗРАБОТКА КОДА ИНФОРМАЦИОННЫХ СИСТЕМ»

## 1.1. Место дисциплины в структуре основной профессиональной образовательной программы

Учебная дисциплина «Разработка кода информационных систем» принадлежит к междисциплинарному циклу.

### Перечень общих компетенций

ОК.01. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам
ОК.02. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности
ОК.04. Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами.
ОК.09. Использовать информационные технологии в профессиональной деятельности

### Перечень профессиональных компетенций

ПК 5.1. Собирать исходные данные для разработки проектной документации на информационную систему.
ПК 5.2. Разрабатывать проектную документацию на разработку информационной системы в соответствии с требованиями заказчика.
ПК 5.3. Разрабатывать подсистемы безопасности информационной системы в соответствии с техническим заданием.
ПК 5.4. Производить разработку модулей информационной системы в соответствии с техническим заданием.
ПК 5.5. Осуществлять тестирование информационной системы на этапе опытной эксплуатации с фиксацией выявленных ошибок кодирования в разрабатываемых модулях информационной системы.

## 1.2. Цель и планируемые результаты освоения дисциплины:

<i><b>Код ОК, ПК</b></i>	<i><b>Умения</b></i>	<i><b>Знания</b></i>
ОК 01, ОК 02, ОК 04, ОК 9, ПК5.1, ПК 5.2, ПК 5.3, ПК 5.4	Осуществлять постановку задачи по обработке информации. Выполнять анализ предметной области. Использовать алгоритмы обработки информации для различных приложений. Работать с инструментальными средствами обработки информации. Осуществлять математическую и информационную постановку задач по обработке информации. Использовать алгоритмы обработки информации для различных приложений.	Основные платформы для создания, исполнения и управления информационной системой. Методы и средства проектирования информационных систем. Основные понятия системного анализа Объектно-ориентированное программирование. Спецификации языка программирования, принципы создания графического пользовательского интерфейса

	Создавать и управлять проектом по разработке приложения и формулировать его задачи. Использовать языки структурного, объектно-ориентированного программирования для создания независимых программ. Использовать методы тестирования в соответствии с техническим заданием	(GUI), файлового ввода-вывода
--	---	-------------------------------

## 2. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

### 2.1. Объем учебной дисциплины и виды учебной работы

Вид учебной работы	Объем в часах
<b>Объем образовательной программы</b>	<b>134</b>
в том числе:	
теоретическое обучение	54
практические занятия	54
Самостоятельная работа	20
<b>Консультации</b>	<b>6</b>
<b>Промежуточная аттестация</b>	<b>экзамен</b>

## 2.2. Тематический план и содержание учебной дисциплины «МДК.05.02 Разработка кода информационных систем»

Наименование разделов и тем	Содержание учебного материала и формы организации деятельности обучающихся	Объем в часах	Коды компетенции, формированию которых способствует элемент программы
<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>
<b>Тема 1. Основные инструменты для создания, исполнения и управления информационной системой</b>	<b>Содержание учебного материала</b>	<b>2</b>	ОК 1, ОК 2, ОК 4, ОК 9, ПК 5.1, ПК 5.2
	Роль анализа предметной области в процессе разработки программного обеспечения. Основные понятия объектно-ориентированного анализа. Язык UML. Использование UML на разных этапах разработки ПО. Инструментальные средства объектно-ориентированного анализа и программирования. Обзор современных CASE средств, поддерживающих UML. Связь пакетов и этапов проектирования.		
	Функциональная модель системы. Создание диаграммы прецедентов использования. Диаграммы классов. Понятие класса. Стереотипы классов. Типы связей. Пакеты. Принципы распределение классов по пакетам. Обнаружение классов		
	<b>В том числе практических занятий и лабораторных работ</b>	<b>12</b>	
	<b>Самостоятельная работа обучающихся</b>	<b>36</b>	
<b>Тема 2. Инструментальная среда MS Visual Studio</b>	<b>Содержание учебного материала</b>	<b>4</b>	ОК 1, ОК 2, ОК 4 ОК 9, ПК 11.1-11.5, ПК 5.3-5.5
	Понятие инструментальной среды разработки. Состав инструментальной среды разработки. Объектно–ориентированные языки программирования. Структура программы. Описание свойств и методов. Оператор вызова метода. Формальные и фактические параметры. Область видимости объектов, методов и свойств. Функции и процедуры объектов. Рекурсивные вызовы. Настройка среды разработки. Проекты и приложения. Файлы и папки проектов. Компоненты проекта. Компоновка и запуск приложения. Создание Windows Form приложения.		
	Простейшие типы и выражения. Операторы ветвления, цикла. Объекты, используемые в приложении. Настройка свойств формы. Элементы		

	управления, события, свойства, методы		
	Типы ошибок. Синтаксические ошибки. Средства отладки приложений. Контрольные точки. Проверка значений переменных. Средства пошагового выполнения программы. Обработка исключительных ситуаций. Структурная обработка исключений.		
	Группирующие элементы управления. Групповая панель с заголовком - GroupBox. Простая панель – Panel. Вкладки TabControl. ЭУ таймер. Строка состояния, шкала, индикатор. Использование ЭУ Панель SplitControl. Полосы прокрутки. Создание главного и контекстного меню		
	<b>В том числе практических занятий и лабораторных работ</b>	<b>16</b>	
	<b>Самостоятельная работа обучающихся</b>	<b>30</b>	
<b>Тема 3. Объектно-ориентированное программирование</b>	<b>Содержание учебного материала</b>	<b>4</b>	ОК 1, ОК 2, ОК 4 ОК 9, ПК 11.1-11.5, ПК 5.3-5.5
	Основные принципы объектно-ориентированного программирования. Наследования. Полиморфизм. Инкапсуляция. Абстракция. Понятие класса. Описание класса. Интерфейсная, исполняемая и инициализирующая части.		
	Создание класса. Использование схемы классов. Инструменты схемы классов. Члены класса. Поля класса. Закрытые, защищенные и открытые поля. Константы. Создание метода класса. Статический и простой метод класса. Конструктор. Наследование класса. Создание класса наследника на основе базового класса. Создание свойств класса. Создание события класса. События и делегаты. Процедуры и функции – методы класса.		
	Создание приложения на основе классов. Структура проекта. Основные элементы проекта. Главный модуль. Задание стартовой формы. Модуль формы. Объявление класса формы. Компиляция и построение проекта. Построение решений и проектов. Перестройка решений и проектов. Очистка решений и проектов.		
	Основные операции с файлами. Чтение и запись файла. Класс FileStream. Считывание данных из текстового файла. Запись данных из текстового файла. Открытие и создание файла для чтения и записи. Обзор основных интерфейсов приложений Windows Forms. Однооконный интерфейс. Многооконный интерфейс. Примеры многооконного интерфейса. Главная форма. Дочерняя форма. Свойства формы. Использование стандартных диалоговых окон. Диалоговые окна для работы с файлами. Работа с буфером обмена		

	<b>В том числе практических занятий и лабораторных работ</b>	<b>14</b>	
	<b>Самостоятельная работа обучающихся</b>	<b>30</b>	
	<b>Консультации</b>	<b>4</b>	
	<b>Промежуточная аттестация: зачёт</b>		
<b>Примерная тематика практических занятий:</b> Объектно-ориентированный анализ Создание проекта VB в инструментальной среде MS Visual Studio Создание проекта C# в инструментальной среде MS Visual Studio Разработка целостного приложения Microsoft Windows Forms Изучение режимов и инструментов отладки программ Создание классов Работа с файлами Создание MDI-приложений		<b>42</b>	
<b>Всего:</b>		<b>134</b>	



### 3. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ

#### 3.1. Для реализации программы учебной дисциплины должны быть предусмотрены следующие специальные помещения:

Лаборатории «Организации и принципов построения информационных систем» оснащенная следующим оборудованием и техническими средствами обучения:

- Автоматизированные рабочие места на 12-15 обучающихся (процессор не ниже Core i3, оперативная память объемом не менее 8 Гб) или аналоги;
- Автоматизированное рабочее место преподавателя (процессор не ниже Core i3, оперативная память объемом не менее 8 Гб) или аналоги;
- Проектор и экран;
- Маркерная доска;
- Программное обеспечение общего и профессионального назначения, в том числе включающее в себя следующее ПО:  
EclipseIDEforJavaEEDevelopers, .NETFrameworkJDK 8, MicrosoftSQLServerExpressEdition, MicrosoftVisioProfessional, MicrosoftVisualStudio, MySQLInstallerforWindows, NetBeans, SQLServerManagementStudio, MicrosoftSQLServerJavaConnector, AndroidStudio, IntelliJIDEA.

#### 3.2. Информационное обеспечение реализации программы

Для реализации программы библиотечный фонд образовательной организации должен иметь печатные и/или электронные образовательные и информационные ресурсы, рекомендуемых для использования в образовательном процессе

##### 3.2.1. Электронные издания (электронные ресурсы)

###### Основная литература

Гниденко, И. Г. Технология разработки программного обеспечения : учебное пособие для среднего профессионального образования / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — Москва : Издательство Юрайт, 2020. — 235 с. — URL: <https://urait.ru/bcode/453640>

###### Дополнительная литература.

Соколова, В. В. Разработка мобильных приложений : учебное пособие для среднего профессионального образования / В. В. Соколова. — Москва : Издательство Юрайт, 2020. — 175 с. — (Профессиональное образование). — ISBN 978-5-534-10680-0. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/456795>

### 4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ

#### 4.1. Критерии, формы и методы оценки результатов обучения

Результаты обучения	Критерии оценки	Формы и методы
---------------------	-----------------	----------------

		<b>оценки</b>
<p><b>Перечень знаний, осваиваемых в рамках дисциплины</b></p> <p>Основные платформы для создания, исполнения и управления информационной системой.</p> <p>Основные платформы для создания, исполнения и управления информационной системой.</p> <p>Методы и средства проектирования информационных систем.</p> <p>Основные понятия системного анализа</p> <p>Объектно-ориентированное программирование.</p> <p>Спецификации языка программирования, принципы создания графического пользовательского интерфейса (GUI), файлового ввода-вывода</p>	<p>«Отлично» - теоретическое содержание курса освоено полностью, без пробелов, умения сформированы, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко.</p> <p>«Хорошо» - теоретическое содержание курса освоено полностью, без пробелов, некоторые умения сформированы недостаточно, все предусмотренные программой учебные задания выполнены, некоторые виды заданий выполнены с ошибками.</p> <p>«Удовлетворительно» - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые умения работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки.</p> <p>«Неудовлетворительно» - теоретическое содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки.</p>	<p>Примеры форм и методов контроля и оценки</p> <ul style="list-style-type: none"> <li>– Компьютерное тестирование по теме</li> <li>– Практическая работа</li> <li>– Устный опрос</li> </ul>
<p><b>Перечень умений, осваиваемых в рамках дисциплины</b></p> <p>Осуществлять постановку задачи по обработке информации.</p> <p>Выполнять анализ предметной области.</p> <p>Использовать алгоритмы обработки информации для различных приложений.</p> <p>Работать с инструментальными средствами обработки информации.</p> <p>Осуществлять математическую и информационную постановку задач по обработке информации.</p> <p>Использовать алгоритмы обработки информации для различных приложений.</p> <p>Создавать и управлять проектом по разработке приложения и формулировать его задачи.</p> <p>Использовать языки структурного, объектно-ориентированного программирования для создания независимых программ.</p> <p>Использовать методы</p>		

тестирования в соответствии с техническим заданием		
--	--	--

#### 4.2. Формы и методы текущего контроля успеваемости обучающихся и промежуточной аттестации

Формы текущего контроля успеваемости:

**Устный опрос (УО)** позволяет выявить правильность ответа по содержанию, его последовательность, самостоятельность суждений и выводов, степень развития логического мышления

**Тестирование (Т)** – задания, с вариантами ответов.

Критерии оценивания

Оценки «отлично» заслуживает студент, если он ответил правильно на 90% вопросов теста

Оценки «хорошо» заслуживает студент, если он ответил правильно на часть вопросов 75%-90%;

Оценки «удовлетворительно» заслуживает студент, если он правильно ответил часть вопросов 50%-75%;

Оценки «неудовлетворительно» заслуживает студент, если он правильно ответил менее чем на 50% вопросов.

**Практическое задание (ПЗ)** используется для закрепления теоретических знаний и отработки навыков и умений, способности применять знания при решении конкретных задач.

Критерии оценивания

Оценки «отлично» заслуживает студент, обнаруживший глубокое знание материала, умение свободно выполнять задания, понимающий взаимосвязь основных понятий темы;

Оценки «хорошо» заслуживает студент, обнаруживший полное знание материала; успешно выполняющий предусмотренные задания; и допустивший незначительные ошибки: неточность фактов, стилистические ошибки;

Оценки «удовлетворительно» заслуживает студент, обнаруживший знания основного материала в объеме, необходимом для дальнейшего изучения дисциплины. Справляющийся с выполнением заданий; допустивший погрешности в ответе, но обладающий необходимыми знаниями для их устранения под руководством преподавателя;

Оценки «неудовлетворительно» заслуживает студент, обнаруживший существенные пробелы в знании основного материала; не справляющийся с выполнением заданий, допустивший серьезные погрешности в ответах, нуждающийся в повторении основных разделов курса под руководством преподавателя.

Формы текущего контроля

	Название темы	Формы текущего контроля успеваемости, промежуточной аттестации
1	Тема 1. Основные инструменты для создания, исполнения и управления информационной системой	Т, ПЗ, УО
2	Тема 2. Инструментальная среда MS Visual Studio	Т, ПЗ, УО
3	Тема 3. Объектно-ориентированное программирование	Т, ПЗ, УО

Примечание. В столбце «Форма текущего контроля успеваемости, промежуточной аттестации» перечисляются все используемые в учебном процессе по данной дисциплине формы контроля освоения материала. (Т – тестирование; ПЗ– практическое задание, УО- устный опрос).

#### **4.3 Материалы текущего и промежуточного контроля успеваемости обучающихся**

##### **Примеры типовых заданий для практических работ**

#### **Тема 1. Основные инструменты для создания, исполнения и управления информационной системой**

##### **Практическая работа 1. Объектно-ориентированный анализ**

1. Краткий обзор и назначение диаграмм.
2. Инструментальные средства объектно-ориентированного анализа.
3. Функциональная модель системы.
4. Создание диаграммы прецедентов использования (Use case диаграмм).
5. Создание диаграммы классов.
6. Понятие класса.
7. Атрибуты и методы класса.
8. Связи

#### **Тема 2. Инструментальная среда MS Visual Studio**

##### **Практическая работа 2. Создание проекта VB в инструментальной среде MS Visual Studio**

1. Понятие инструментальной среды разработки.
2. Состав инструментальной среды разработки.
3. Языки программирования высокого уровня.
4. Обзор основных интерфейсов приложений Windows Forms.
5. Структура программы.
6. Проекты и приложения.
7. Файлы и папки проектов.
8. Компоненты проекта.
9. Компоновка и запуск приложения.
10. Окно редактора кода. Объекты, используемые в приложении.
11. Задание стартовой формы.
12. Настройка свойств формы.
13. Элементы управления, события, свойства, методы.
15. Создание исполняемого файла (exe файл).
16. Тестирование созданного приложения

### **Практическая работа 3. Создание проекта C# в инструментальной среде MS Visual Studio**

1. Структура программы.
2. Проекты и приложения.
3. Файлы и папки проектов.
4. Компоненты проекта.
5. Окно редактора кода. Объекты, используемые в приложении.
6. Простейшие типы и выражения.
7. Объявления переменных.
8. Константы.
9. Вставка комментариев.
10. Ветвления. Конструкции IF ...Else, Switch.
11. Циклы: For, For Each, While, Do.
12. Использование финансовых функций Excel.
13. Вызов функции Visual Basic.
14. Использование функции TryParse.
15. Открытие и закрытие форм.
16. Открытие формы в модальном режиме.
17. Задание стартовой формы.
18. Компоновка и запуск приложения.

### **Практическая работа 4. Разработка целостного приложения Microsoft Windows Forms**

1. Основные элементы управления.
2. Общие свойства элементов управления.
3. Элементы управления для работы с текстом.
4. Проверка вводимых данных с помощью регулярных выражений.
5. Элементы выбора.
6. Группирующие элементы управления.
7. Использование в форме графики.
8. Полосы прокрутки.
9. ЭУ таймер.
10. Строка состояния.
11. Создание главного и контекстного меню.
12. Создание панели элементов.

### **Практическая работа 5. Изучение режимов и инструментов отладки программ**

1. Типы ошибок.
2. Средства отладки приложений.
3. Контрольные точки.
4. Проверка значений переменных.
5. Средства пошагового выполнения программы.
6. Инструменты для отладки программы.
7. Обработка исключительных ситуаций.
8. Структурная обработка исключений.

### **Тема 3. Основы объектно-ориентированного программирования**

#### **Практическая работа 6. Создание классов**

1. Основные принципы объектно ориентированного программирования.
2. Создание класса.
3. Использование схемы классов.
4. Инструменты схемы классов.
5. Члены класса.

6. Поля класса.
7. Методы класса.
8. Создание класса наследника на основе базового класса.
9. Создание свойств класса.
10. Создание события класса.
11. События и делегаты.
12. Создание приложения на основе классов.

#### **Практическая работа 7. Работа с файлами**

1. Основные операции с файлами.
2. Чтение и запись файла.
3. Класс FileStream.
4. Считывание данных из текстового файла.
5. Запись данных из текстового файла.
6. Открытие и создание файла для чтения и записи.

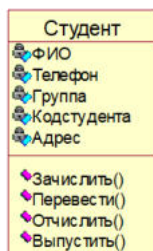
#### **Практическая работа 8. Создание MDI-приложений**

1. Обзор основных интерфейсов приложений WindowsForms.
2. Однооконный интерфейс.
3. Многооконный интерфейс MDI (Multiple Document Interface).
4. Примеры многооконного интерфейса.
5. Главная форма. Свойства формы.
6. Дочерняя форма.
7. Отображение дочерней формы.
8. Расположение дочерних форм.
9. Использование стандартных диалоговых окон.
10. Диалоговые окна для работы с файлами.
11. Работа с буфером обмена

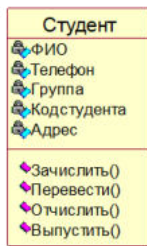
#### **Примеры тестовых заданий**

#### **Тема 1. Основные инструменты для создания, исполнения и управления информационной системой**

##### **1) Что изображено на рисунке**



- a) Состояние
  - b) Класс
  - c) Активность
  - d) Пакет
- 2) Группа - это



- a) Состояние
  - b) Класс
  - c) Атрибут
  - d) Операция
- 3) \_\_\_\_\_ представляют собой процессы, реализуемые некоторым классом
- a) Атрибуты
  - b) Операции
  - c) состояние
- 4) \_\_\_\_\_ - это элемент информации, связанный с классом.
- a) Атрибут
  - b) Операция
  - c) состояние
- 5) Какой вид связи изображен на рисунке



- a) Ассоциация
  - b) Обобщение
  - c) Агрегация
  - d) Композиция
- 6) Какой вид связи изображен на рисунке



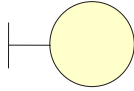
- a) Ассоциация
  - b) Обобщение
  - c) Агрегация
  - d) Композиция
- 7) Какой вид связи изображен на рисунке



- a) Ассоциация
  - b) Обобщение
  - c) Агрегация
  - d) Композиция
- 8) Связь между целым и его частью.

- a) Ассоциация
- b) Обобщение
- c) Агрегация

9) Какой стереотип класса изображен на рисунке



Страница регистрации

- a) Граничный класс
- b) Класс сущность
- c) Управляющий класс

10) Какой стереотип класса изображен на рисунке



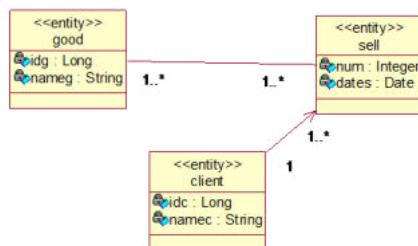
Список клиентов

- a) Граничный класс
- b) Класс сущность
- c) Управляющий класс

11) Для моделирования статистической структуры объектов и связей между ними используются

- a) диаграммы классов
- b) диаграммы активности
- c) диаграммы вариантов использования
- d) диаграмма последовательности

12) Какая диаграмма представлена на рисунке



- a) диаграммы классов
- b) диаграммы активности
- c) диаграммы вариантов использования
- d) диаграмма последовательности

13) Процессы, реализуемые некоторым классом

- a) Атрибут
- b) Операция
- c) Сообщение
- d) Имя

14) Некоторое свойство моделируемой сущности, которым обладают все объекты данного класса

- a) Атрибут



- b) Операция
  - c) Сообщение
  - d) Имя
- 15) Хранение в одной структуре и данных, и методов называется**
- a) свойством
  - b) полиморфизмом
  - c) наследованием
  - d) инкапсуляцией
- 16) Упорядочение абстракций, расположение их по уровням**
- a) наследование
  - b) инкапсуляция
  - c) модульность
  - d) иерархия
- 17) Скрытие информации, т.е. маскировка всех внутренних деталей, не влияющих на внешнее поведение**
- a) наследование
  - b) абстрагирование
  - c) полиморфизм
  - d) инкапсуляция
- 18) Выделение существенных характеристик объекта, которые отличают его от всех других видов объектов и таким образом чётко определяются особенности данного объекта с точки зрения дальнейшего его рассмотрения**
- a) полиморфизм
  - b) инкапсуляция
  - c) абстрагирование
  - d) наследование
- 19) Как называется методология, при которой требования формируются на основе понятий классов и объектов, составляющих словарь предметной области**
- a) объектно-ориентированный анализ
  - b) функциональный анализ
  - c) структурный анализ
  - d) предметный анализ
- 20) Связь между целым и его частью**
- a) Агрегация
  - b) Ассоциация
  - c) Обобщение
  - d) Параметризация
- 21) Как называется структурное отношение, показывающее, что объекты одного типа связаны с объектами другого типа**
- a) Агрегация
  - b) Обобщение
  - c) Ассоциация
  - d) Композиция
- 22) Как называется отношение типа «является частью» («is-part-of»), когда объект-целое состоит из нескольких объектов-частей и когда объект-часть может существовать без объекта-целого**
- a) Агрегация
  - b) Обобщение
  - c) Ассоциация
  - d) Композиция

- 23) Как называется отношение, когда время жизни частей и целого совпадают и, когда объект-часть не может существовать без объекта-целого
- Агрегация
  - Обобщение
  - Ассоциация
  - Композиция

1	2	3	4	5	6	7	8	9	10	11	12
b	b	b	a	a	b	c	c	a	b	a	a
13	14	15	16	17	18	19	20	21	22	23	
b	a	c	d	d	c	a	a	c	a	d	

## Тема 2. Инструментальная среда MS Visual Studio

- Ключевое слово \_\_\_\_ сообщает компилятору о необходимости определения типа переменной из выражения, находящегося с правой стороны оператора (C#)
  - float
  - char
  - variant
  - var
- Правильное описание константы(C#)
  - float const Pi = 3.14;
  - char const Pi = 3.14;
  - const char Pi = 3.14;
  - const float Pi = 3.14;
- Данная инструкция Label1.Text=String.Format("{0:C}",T) позволяет вывести результат в формате
  - даты
  - валюты
  - десятичного числа
  -
- Объявление переменной. Строка символов.(C#)
  - float a;
  - string a;
  - char a;
  - int a;
- Объявление переменной. Число с плавающей точкой.(C#)
  - float a;
  - string a;
  - char a;
  - int a;
- ::Синт6::Объявление переменной. Целое число.(C#)
  - float a;
  - string a;
  - char a;
  - int a;
- Сравнение.Проверка на равенство(C#)
  - a=b
  - a==b
  - a<>b

- d)  $a!=b$
- 8) **Сравнение. Проверка на неравенство(C#)**
- a)  $a=b$
  - b)  $a==b$
  - c)  $a<>b$
  - d)  $a!=b$
- 9) **Что обозначает  $a!=b$ (C#)**
- a) переменную a увеличить на b
  - b) проверка на равенство
  - c) проверка на неравенство
  - d) переменную a уменьшить на b
- 10) **Что обозначает  $a==b$ (C#)**
- a) переменную a увеличить на b
  - b) проверка на равенство
  - c) проверка на неравенство
  - d) переменную a уменьшить на b
- 11) **Что обозначает  $a+=b$ (C#)**
- a) переменную a увеличить на b
  - b) проверка на равенство
  - c) проверка на неравенство
  - d) переменную a уменьшить на b
- 12) **Что обозначает  $a-=b$ (C#)**
- a) переменную a увеличить на b
  - b) проверка на равенство
  - c) проверка на неравенство
  - d) переменную a уменьшить на b
- 13) **Что обозначает  $a++$ (C#)**
- a) переменную a увеличить на 1
  - b) проверка на равенство 1
  - c) проверка на неравенство 1
- 14) **Инструкция, сообщающая программе о необходимости выполнения одного из множества условий (C#)**
- a) switch
  - b) select
  - c) if
  - d) do
- 15) **Выберите правильную инструкцию(C#)**
- a) for (int i=0; i<5; i++)
  - b) for (int i=0, i<5, i++)
  - c) for (int i=0 i<5 i++)
  - d) for (int i=0: i<5: i++)
- 16) **Какой цикл используется, когда известно количество итераций**
- a) for
  - b) do
  - c) while
- 17) **Какой цикл используется, когда известно количество итераций**
- a) for
  - b) do
  - c) while

- 18) Какой цикл позволяет выполнять тело цикла до тех пор, пока указанное условие продолжает оставаться истинным
- for
  - do
  - while
- 19) Какой цикл удобно использовать при переборе элементов массива
- for
  - do
  - while
  - for each
- 20) В языке VB существует массивы массивов (ступенчатый, jagged)
- да
  - нет
- 21) В языке C# существует массивы массивов (ступенчатый, jagged)
- да
  - нет
- 22) В C# все массивы являются производными от базового класса
- System.Array
  - Array
  - System.Data
  - Data.Array
- 23) Описание массива (C#).
- int[] a,b,c;
  - int() a,b,c;
  - int a(),b(),c();
  - int a[],b[],c[];
- 24) ::Массив11::Описание ступенчатого или ломанного массива (C#).
- тип [][] имя массива = new тип[размер] [];
  - тип[,] имя = new тип [ разм\_1, разм\_2 ];
  - тип[] имя = new тип [ размерность ];
- 25) ::Массив12::Описание двумерного массива (C#).
- тип [][] имя массива = new тип[размер] [];
  - тип[,] имя = new тип [ разм\_1, разм\_2 ];
  - тип[] имя = new тип [ размерность ];
- 26) ::Массив13::Описание одномерного массива (C#).
- тип [][] имя массива = new тип[размер] [];
  - тип[,] имя = new тип [ разм\_1, разм\_2 ];
  - тип[] имя = new тип [ размерность ];
- 27) Какое окно удобно использовать для того чтобы выяснить, какие переменные находятся под влиянием текущего алгоритма
- Локальные
  - Интерпретация
  - Видимые
  - Контрольные значения
- 28) В каком окне отображаются переменные только из текущей и предыдущей строк кода (C#)
- Локальные
  - Интерпретация
  - Видимые
  - Контрольные значения

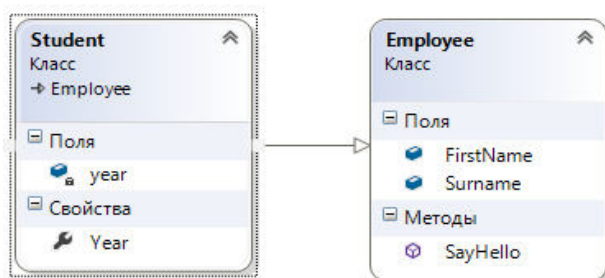
- 29) В каком окне выводятся значения всех переменных, содержащиеся во фрагменте кода, начиная с оператора, расположенного на 3 строки выше контрольной точки. (VB)
- a) Локальные
  - b) Интерпретация
  - c) Видимые
  - d) Контрольные значения
- 30) Какое окно позволяет создавать индивидуальный список переменных
- a) Локальные
  - b) Интерпретация
  - c) ~Видимые
  - d) Контрольные значения
- 31) В каком окне из окон, предназначенных для отладки программы? можно выполнить следующие действия (рисунок1)
- a) Локальные
  - b) Интерпретация
  - c) Видимые
  - d) Контрольные значения
- 32) Если в блоке Try возникает исключение, блок Finally выполняется \_\_\_\_.
- a) До выполнения блока Catch
  - b) Вместо блока Catch
  - c) После выполнения блока Catch
  - d) Ни один вариант не подходит
- 33) Как правило оператор Try содержит \_\_\_\_ блок и хотя бы один блок \_\_\_\_
- a) Catch, Try
  - b) Try, Catch
  - c) Throw, Catch
  - d) Catch, Throw
- 34) Если в блоке Try исключений не было , приложение игнорирует \_\_\_\_ соответствующего оператора Try.
- a) Блоки Catch
  - b) Оператор Return
  - c) Подходят оба предыдущих оператора
  - d) Ни один вариант не подходит
- 35) Чтобы сообщить о выполнении исключительных ситуаций в программе, можно использовать оператор . (C#)
- a) Оператор Return
  - b) throw
  - c) exception
  - d) finally
  - e) error
- 36) Правильная инструкция для обработки исключения на C#
- a) Catch Ex AS Exception
  - b) catch (Exception ex)
  - c) Catch When Err.Number=13
- 37) Правильная инструкция для обработки исключения на VB
- a) Catch Ex AS Exception
  - b) catch (Exception ex)
  - c) throw new Exception

- 38) Что в Visual Studio позволяет добавлять элементы управления к форме с помощью визуальной технологии.
- Обозреватель решений
  - Стандартная
  - Панель Элементов
  - Обозреватель решений
- 39) Какое окно в Visual Studio позволяет просматривать файлы решения.
- Командный обозреватель
  - Обозреватель решений
  - Обозреватель объектов
  - Обозреватель серверов
- 40) Где в Visual Studio задается начальная форма (C#).
- в файле Program.vb
  - в файле Program.cs
  - В окне свойств объекта
  - В окне свойств проекта

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
d	d	b	b	a	d	b	d	a	b	a	d	a	a	a
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
a	b	c	c	b	a	a	a	a	b	c	a	c	c	d
31	32	33	34	35	36	37	38	39	40					
b	c	b	a	b	b	a	c	b	b					

### Тема 3. Объектно-ориентированное программирование

- 1) В объектно-ориентированном программировании \_\_ \_\_ \_\_ \_\_ представляют собой типы сходных объектов и в определенном смысле аналогичны строительным чертежам
- Классы
  - Атрибуты
  - Свойства
  - Модели поведения
- 2) Свойство элемента управления CommandButton , позволяющее отобразить текст на кнопке
- Value
  - Name
  - Text
  - Caption
- 3) Какой класс является дочерним



- a) Employer
  - b) Student
- 4) Доступ к классу возможен из любого места одной сборки либо из другой сборки, на которую есть ссылка
- a) public
  - b) internal
- 5) \_\_\_\_\_ – это переменная, объявленная внутри класса
- a) Поле
  - b) Свойство
  - c) Событие
  - d) Метод
- 6) Объекты только этого класса могут обращаться к данному полю
- a) Private
  - b) Public
  - c) Protected
- 7) Объекты любого класса могут обращаться к этому полю
- a) Private
  - b) Public
  - c) Protected
- 8) Только объекты классов-наследников могут обращаться к полю
- a) Private
  - b) Public
  - c) Protected
- 9) Что представлено на рисунке

```

public void SayHello(out string fullname)
{
    // throw new System.NotImplementedException();
    fullname = "Hello, " + FirstName + " " + Surname;
    return;
}

```

- a) Свойство
  - b) Событие
  - c) Поле
  - d) Метод
- 10) Что представлено на рисунке
- ```

public string FirstName;
public string Surname;

```
- a) Свойство
  - b) Событие
  - c) Поле
  - d) Метод
- 11) \_\_\_\_\_ – это метод класса, предназначенный для инициализации объекта при его создании.

- a) Конструктор
  - b) Деструктор
- 12) В программировании \_\_\_\_\_ позволяет создавать новый класс на базе другого.
- a) Наследование
  - b) Инкапсуляция
  - c) Полиморфизм
  - d) Абстрагирование
- 13) Что представлено во фрагменте кода

```
public class Cashier : Employee
{
```

- a) Класс Casher является дочерним от Employer
  - b) Класс Employer является дочерним от Casher
  - c) Создание экземпляра класса
- 14) Что представлено во фрагменте кода

```
Employee emp = new Employee();
```

- a) Класс Casher является дочерним от Employer
  - b) Класс Employer является дочерним от Casher
  - c) Создание экземпляра класса
- 15) Что представлено во фрагменте кода

```
public int Year
{
    get// аксессор чтения поля
    {
        //throw new System.NotImplementedException();
        return year;
    }
    set// аксессор записи в поле
    {
```

- a) Объявление поля
  - b) Объявление свойства
  - c) Объявление метода
  - d) Объявление события
- 16) Применение \_\_\_\_\_ позволяет объектам реагировать на те или иные ситуации и выполнять необходимые ответные действия.
- a) событий
  - b) полей
  - c) методов
  - d) свойств
- 17) Какое утверждение верно

- a) Событие может вызываться только в том классе, где оно объявлено
  - b) Событие может вызываться любом классе, где оно объявлено
  - c) Событие может вызываться только в дочернем классе
- 18) Что представлено во фрагменте кода

```
public event EventHandler AgeChange;
```



- a) Объявление поля
  - b) Объявление свойства
  - c) Объявление метода
  - d) Объявление события
- 19) Что позволяет скрывать внутреннюю реализацию
- a) Инкапсуляция
  - b) Наследование
  - c) Полиморфизм
  - d) Абстракция
- 20) Способность объектов с одним интерфейсом иметь различную реализацию.
- a) Инкапсуляция
  - b) Наследование
  - c) Полиморфизм
  - d) Абстракция
- 21) Позволяет выделять из некоторой сущности только необходимые характеристики и методы, которые в полной мере (для поставленной задачи) описывают объект.
- a) Инкапсуляция
  - b) Наследование
  - c) Полиморфизм
  - d) Абстракция
- 22) Класс, позволяющий записывать строки в текстовый файл
- a) StreamWrite
  - b) StreamReader
- 23) Класс, позволяющий считывать строки из текстового файла
- a) StreamWrite
  - b) StreamReader
- 24) Метод класса, предназначенный для инициализации объекта при его создании
- a) Построитель
  - b) Деструктор
  - c) Конструктор
  - d) Инициализатор
- 25) Какое утверждение верно
- a) Событие может вызываться любым классе, где оно объявлено
  - b) Событие может вызываться только в дочернем классе
  - c) Событие может вызываться только в том классе, где оно объявлено
- 26) Метод, который имеет доступ к данным объекта, и его вызов выполняется через объект
- a) Статический
  - b) Простой
  - c) Конструктор
  - d) Деструктор
- 27) Внутри тела \_\_\_\_\_ используются специальные процедуры аксессоры (accessors) –Get и Set.
- a) события
  - b) метода
  - c) свойства
  - d) поля
- 28) Метод, который не имеет доступа к полям объекта, и для вызова такого метода не нужно создавать экземпляр (объект) класса, в котором он объявлен

- a) Статический
- b) Деструктор
- c) Конструктор
- d) Простой

**29) Что обозначает модификатор доступа internal**

- a) Доступ возможен из этого класса и из классов, унаследованных от него
- b) Доступ ограничен только текущей сборкой
- c) Доступ не ограничен
- d) Доступ возможен только из этого класса

**30) Что обозначает модификатор доступа protected**

- a) Доступ возможен из этого класса и из классов, унаследованных от него
- b) Доступ ограничен только текущей сборкой
- c) Доступ не ограничен
- d) Доступ возможен только из этого класса

|    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 |
| a  | c  | b  | a  | a  | a  | b  | c  | d  | c  | a  | a  | a  | c  | b  |
| 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 |
| a  | a  | b  | a  | c  | d  | a  | b  | c  | c  | b  | c  | a  | b  | a  |

**Вопросы для подготовки к экзамену:**

1. Инструментальные средства объектно-ориентированного анализа и программирования.
2. Диаграммы классов.
3. Понятие инструментальной среды разработки.
4. Состав инструментальной среды разработки.
5. Языки программирования высокого уровня.
6. Обзор основных интерфейсов приложений Windows Forms.
7. Структура программы.
8. Проекты и приложения.
9. Файлы и папки проектов.
10. Компоненты проекта.
11. Объектно – ориентированные языки программирования VB , C#.
12. Простейшие типы и выражения.
13. Ветвления. Конструкции IF ...Else.
14. Ветвления. Конструкции Switch/Select.
15. Циклы For.
16. Циклы For Each.
17. Циклы While.
18. Циклы Do.
19. Использование финансовых функций Excel в проектах C#.
20. Вызов функции VisualBasic в проектах C#..
21. Открытие и закрытие форм. Открытие формы в модальном режиме.
22. Элементы управления для работы с текстом.
23. Элементы выбора.
24. Группирующие элементы управления.
25. Использование в форме графики.
26. Строка состояния. Таймер.
27. Создание панели элементов и меню

28. Типы ошибок.
29. Средства отладки приложений.
30. Обработка исключительных ситуаций.
31. Основные принципы объектно ориентированного программирования.
32. Создание класса. Использование схемы классов. Инструменты схемы классов.
33. Поля класса.
34. Методы класса.
35. Создание свойств класса.
36. Создание события класса.
37. Чтение и запись файла. Класс FileStream.
38. Обзор основных интерфейсов приложений WindowsForms.
39. Многооконный интерфейс MDI (Multiple Document Inte

