

Документ подписан простой электронной подписью  
Информация о владельце:  
ФИО: Андрей Драгомжрович Хлутков  
Должность: директор  
Дата подписания: 03.06.2024 10:41:30  
Уникальный программный ключ:  
880f7c07c583b07b775f6604a630281b13ca9fd2

**Федеральное государственное бюджетное образовательное учреждение  
высшего образования**

**РОССИЙСКАЯ АКАДЕМИЯ НАРОДНОГО ХОЗЯЙСТВА и ГОСУДАРСТВЕННОЙ СЛУЖБЫ  
при ПРЕЗИДЕНТЕ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**СЕВЕРО-ЗАПАДНЫЙ ИНСТИТУТ УПРАВЛЕНИЯ**

---

**ФАКУЛЬТЕТ СРЕДНЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ**

Утвержден решением цикловой  
(методической) комиссией по  
специальности

09.02.07 «Информационные  
системы и программирование»

Протокол № 1

от «25» декабря 2022 г.

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ**

**МДК.05.02 Разработка кода информационных систем**

*(индекс, наименование дисциплины (модуля), в соответствии с учебным планом)*

Специальность 09.02.07 «Информационные системы и программирование»

Квалификация

Техник на базе основного общего образования

Форма обучения

очная

Год набора

2022

**Автор–составитель:**

старший преподаватель кафедры бизнес-информатики Лахманова Ирина Евгеньевна

**Заведующий кафедрой** бизнес-информатики, доктор военных наук, профессор Наумов

Владимир Николаевич

*(наименование кафедры) (ученая степень и(или) ученое звание)*

*(Ф.И.О.)*

## **Содержание**

<b>1. Перечень планируемых результатов обучения по дисциплине</b>	<b>4</b>
<b>2. Оценочные средства по дисциплине для текущего контроля.</b>	<b>5</b>
<b>2.1. Текущий контроль.</b>	<b>5</b>
<b>2.2. Промежуточная аттестация</b>	<b>34</b>
<b>3. Описание системы оценивания, шкала оценивания</b>	<b>35</b>
<b>3.1. Показатели и критерии оценивания для текущего контроля.</b>	<b>35</b>
<b>3.2. Показатели и критерии оценивания для промежуточного контроля</b>	<b>36</b>

## 1. Перечень планируемых результатов обучения по дисциплине

1. Перечень планируемых результатов обучения по дисциплине - перечень компетенций с указанием компонентов компетенций дисциплины, как отдельного элемента ОП

Дисциплина «Разработка кода информационных систем» обеспечивает овладение следующими компетенциями:

Перечень общих компетенций

### Перечень профессиональных компетенций

Код компетенции	Наименование компетенции	Код компонента компетенции	Наименование компонента компетенции
ПК 5		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

В результате освоения дисциплины у студентов должны быть сформированы:

<i>Код ОК, ПК</i>	<i>Умения</i>	<i>Знания</i>
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	Основные платформы для создания, исполнения и управления информационной системой. Методы и средства проектирования информационных систем.

	<p>приложений.          Работать с инструментальными средствами обработки информации.          Осуществлять математическую и информационную постановку задач по обработке информации.          Использовать алгоритмы обработки информации для различных приложений.          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</p>	<p>Основные понятия системного анализа          Объектно-ориентированное программирование.          Спецификации языка программирования, принципы создания графического пользовательского интерфейса (GUI), файлового ввода-вывода</p>
--	---	--

## 2. Оценочные средства по дисциплине для текущего контроля.

### 2.1. Текущий контроль.

#### 2.1.1 Компетентностно-ориентированные задания по темам

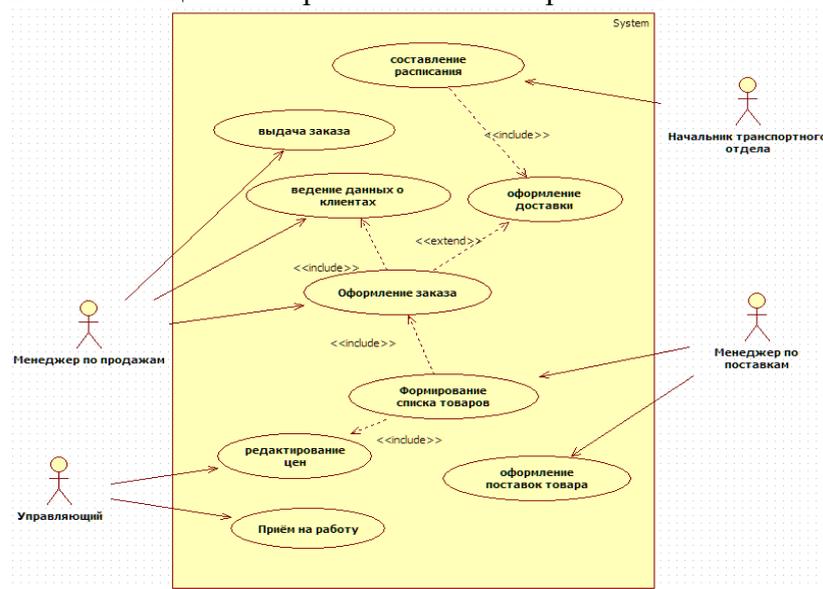
#### Тема 1. Основные инструменты для создания, исполнения и управления информационной системой

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

Задание выполняется в StarUml

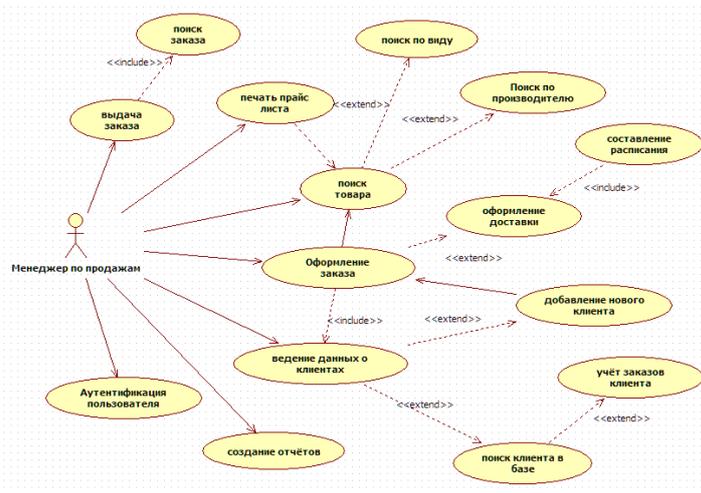
Задание 1

Создать диаграмму прецедентов предметной области. Диаграмма содержит прецеденты, показывающие связь различных категорий пользователей



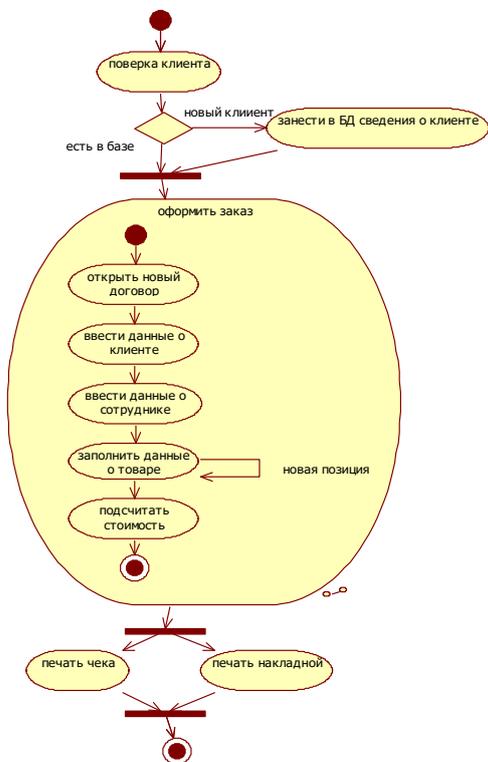
Задание 2

Создать диаграмму прецедентов менеджера по продажам



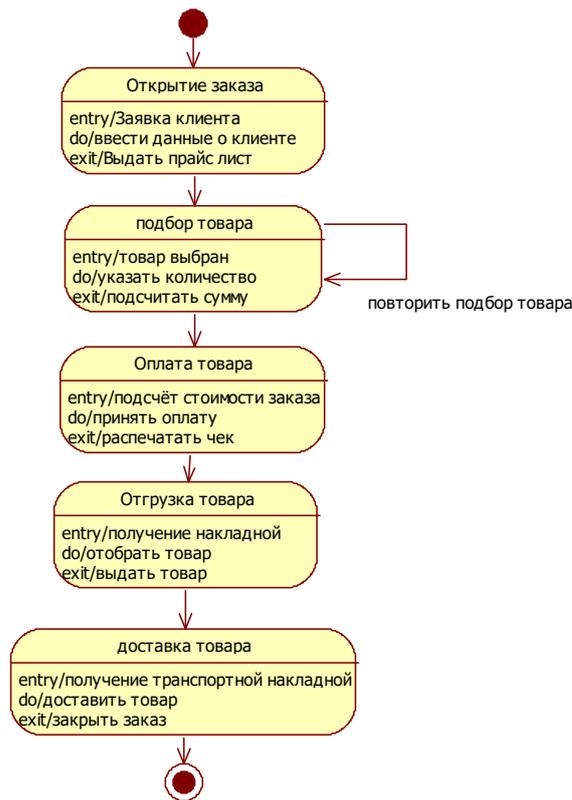
Найти в диаграмме ошибки в использовании ассоциаций. Внести необходимые изменения  
 Аналогичным образом создать диаграммы прецедентов для остальных пользователей  
 Задание 3

Создать диаграмму активности оформления заказа



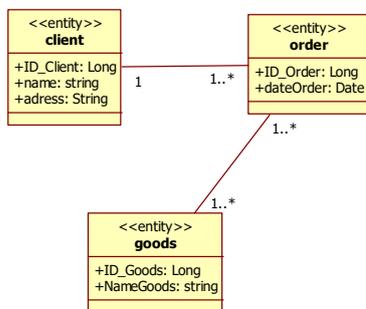
Задание 4.

Создать диаграмму состояния оформление заказа



### Задание 5.

Выявить сущности в рассматриваемой предметной области «Деятельность предприятия торговли» и создать диаграмму классов для  
 На рисунке показан фрагмент диаграммы.



## Тема 2. Инструментальная среда MS Visual Studio

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

Разработать формы программный код для решения следующих задач

#### Задание 1.

Бухгалтерия рассчитывает недельную зарплату сотрудников. В основе расчета лежит почасовая ставка и количество часов, отработанных сотрудником за неделю.

Предполагается, что рабочая неделя составляет 40 часов. Если сотрудник отработал 40 часов и меньше, то зарплата рассчитывается простым умножением часов на ставку.

Переработка свыше 40 часов оплачивается в полуторном размере. Оплата переработки складывается с суммой, начисленной за обычную 40 часовую неделю

Расчет недельной зарплаты

Почасовая ставка: 350

Количество отработанных часов: 45

Начислено: 16 625,00р.

Расчитать

Пример формы

### Задание 2.

Банк предоставляет кредит на срок от 2 до 5 лет (от 24 до 60 месяцев). Заемщики возвращают кредит ежемесячными взносами. Размер платежа зависит от срока и суммы кредита, а также от процентной ставки. Приложение должно позволять вводить стоимость автомобиля, размер первоначального платежа и годовой процент по кредиту. Приложение должно выводить срок кредита в месяцах и размер ежемесячного платежа, при 2-х, 3-х, 4 и 5- летнем кредите. Это позволит пользователю сравнивать планы погашения кредита и выбирать наиболее подходящий.

Расчет кредита на приобретение автомоб...

Цена: 500000

Первый взнос: 100000

Годовой процент: 12

Расчитать

месяцы	ежемесячный платеж
24	18 829,39р.
36	13 285,72р.
48	10 533,53р.
60	8 897,78р.

Пример формы

### Задание 3.

Приложение определяет размер пожертвования после вычета организационных расходов. Часть каждого пожертвования (17%) уходит на обслуживание счета. Приложение должно выводить размер пожертвования, а также отображать общую сумму фонда.

Сбор пожертвований

21.01.2010 Дата: 21.01.2010

Введите сумму: 50000

Внести

Дата	Сумма
20.01.2010	16600
20.01.2010	8300
21.01.2010	12450
21.01.2010	40670
21.01.2010	16600
21.01.2010	41500

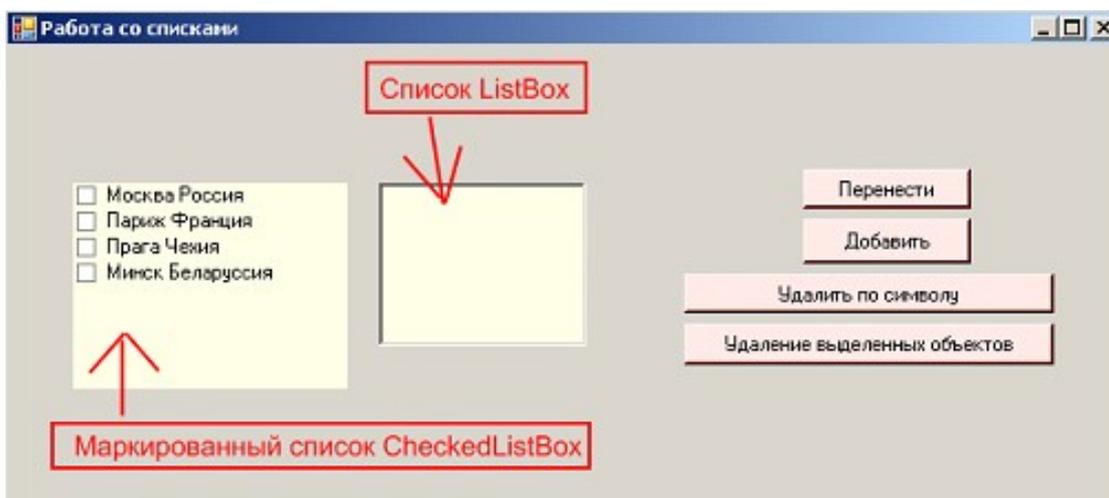
Сумма с учетом налога: 41 500,00р.

Всего: 136 120,00р.

Пример формы

### Задание 4.

Для формы, представленной на рисунке, создать процедуры обработки событий для кнопок: перенести, добавить, удалить, удалить выделенное



Задание 5.

Собрать созданные формы в единое приложение

Выбрать инструмент элемент MenuStrip

Ввести основные пункты

Добавить подпункты

Двойной щелчок по пункту меню позволяет добавить программный код, например:

```
Private Sub КредитToolStripMenuItem_Click(ByVal sender As System.Object,  
ByVal e As System.EventArgs) Handles КредитToolStripMenuItem.Click
```

```
Car_Payment_Calculator.Show()
```

```
End Sub
```

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

Задание 1.

Разработать программу, моделирующую работу коробки передач автомобиля в зависимости от скорости движения ( $V$ ) транспортного средства

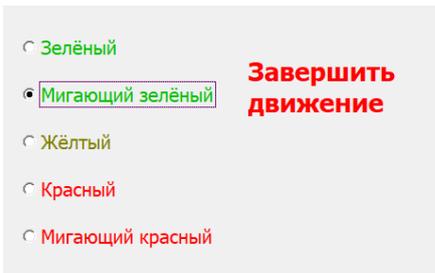
- $V=0$  - нейтральное положение
- $0 < V < 5$  км/час – 1 передача
- $5 \leq V < 20$  км/час – 2 передача
- $20 \leq V < 40$  км/час – 3 передача
- $40 \leq V < 60$  км/час – 4 передача
- $60 \leq V < 80$  км/час – 5 передача
- $V \geq 80$  км/час – 6 передача

Создать форму, расположив на ней ЭУ текстовое поле, счётчик, метку и кнопку.

Процедура обработки событие нажатие кнопки передаёт в текстовое поле значение скорости, выбранной счётчиком и отображает в метке номер передачи

Задание 2.

Создать форму и процедуру обработки события, моделирующую работу светофора. Вид формы представлен на рисунке



### Задание 3.

Составить программу принятия решения.

№ п/п	факт (атрибут)	№ атрибута	характеристика атрибута	весовой фактор атрибута
1	Курс акций в данный момент	1.1	Высокий	50
		1.2	Средний	30
		1.3	Низкий	10
2	Вероятность снижения курса акций	2.1	Высокий	40
		2.2	Средний	20
		2.3	Низкий	10
3	Потребность в наличных деньгах	3.1	Высокий	80
		3.2	Средний	50
		3.3	Низкий	20

Известны:

- курс акций в данный момент
- вероятность снижения
- потребность в деньгах
- и вес каждого параметра в принятии решения.

Требуется: вывести решение о продаже или покупке акций

Правила принятия решения

Если суммарный весовой фактор  $< 100$ , то "Не продавать"

Если суммарный весовой фактор  $\geq 100$  и  $< 140$ , то "Наблюдать"

Если суммарный весовой фактор  $\geq 140$ , то "Продавать"

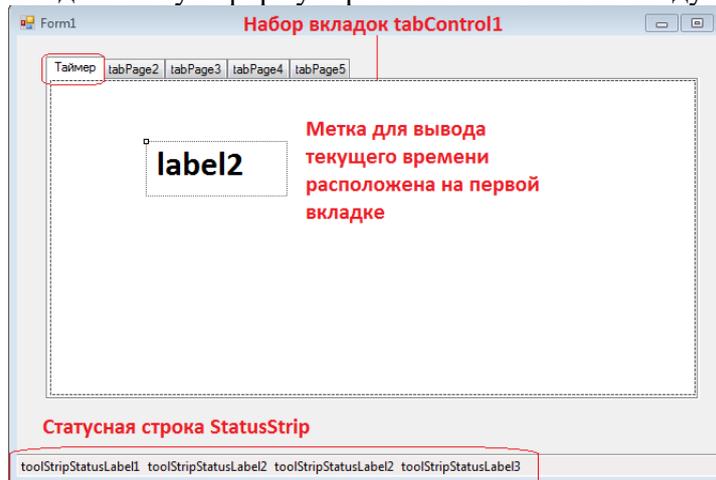
### Практическая работа 4. Разработка целостного приложения Microsoft Windows Forms

Создать новый проект

Задание 1.

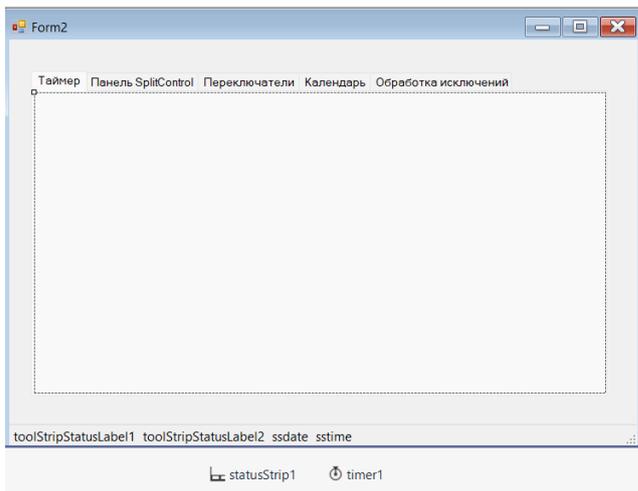
Создать форму с вкладками.

Создать новую форму и расположить на ней следующие ЭУ

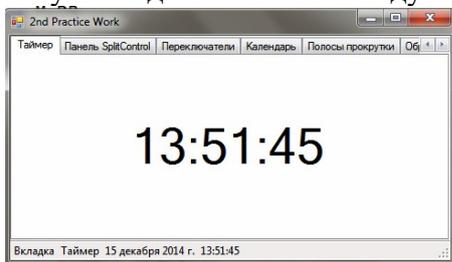


Добавить компонент Timer установить Interval –1000 (соответствует 1 секунде) и строку состояния StatusStrip, которая будет отображать время и текущую вкладку, а также подписи к ним, то есть в статусной строке четыре метки

Изменить подписи вкладок

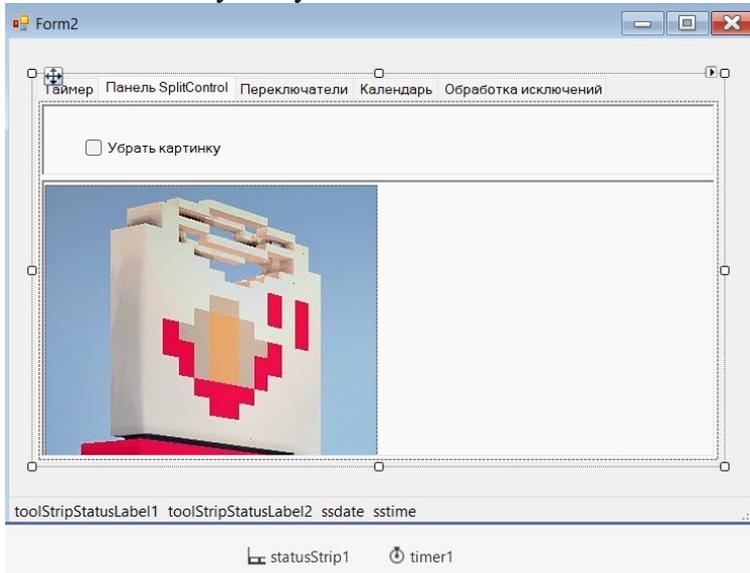


Расположить на форме ЭУ Label, который будет отображать системное время  
 Разработать код инициализации таймера при загрузке формы, вывод системного времени в статусную строку и название открытой вкладки  
 Результат должен иметь следующий вид



Задание 2.

Создать вкладку с двумя изменяемыми областями ЭУ Панель SplitControl



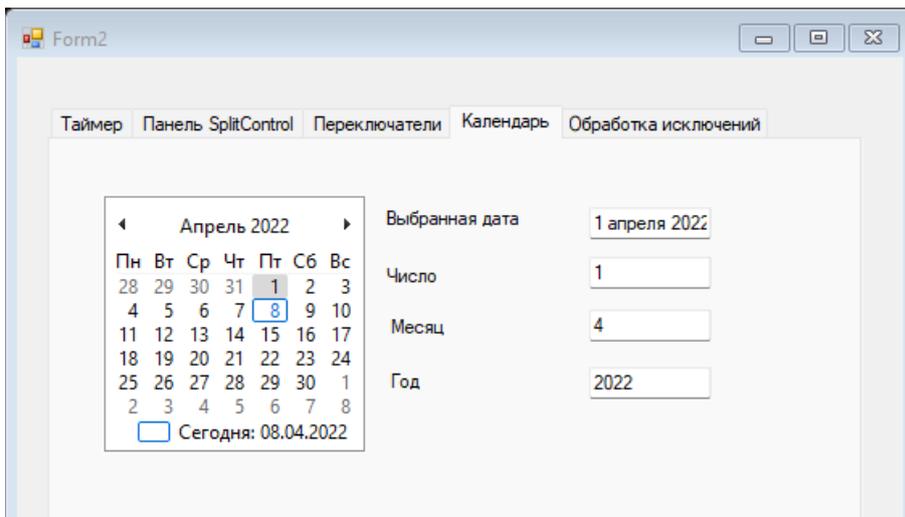
Загрузить на одну область рисунок, а во второй ЭУ  CheckBox, изменение состояния которого определяет отображение рисунка

Разработать программный код, управляющий отображением рисунка

Задание 3.

Разработать интерфейс вкладки календарь, как показано на рисунке. Расположить на вкладке следующие ЭУ:

- календарь 
- четыре текстовых поля для вывода параметров даты

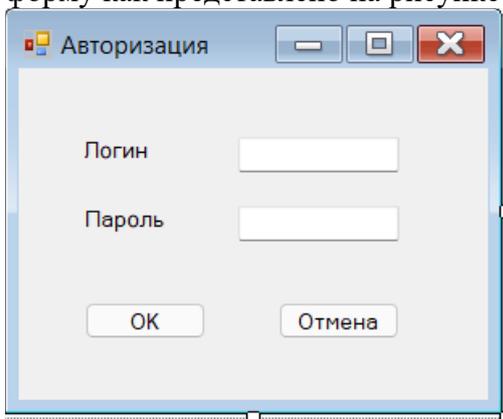


Создать процедуру выбора даты, фрагмент кода которой представлен на рисунке

```
private void monthCalendar1_DateSelected(object sender, DateRangeEventArgs e)
{
    //запись в поля: даты, дня, месяца, и года используя выделенный элемент ЭУ календарь
    textBox1.Text = monthCalendar1.SelectionStart.ToLongDateString();
    textBox2.Text = monthCalendar1.SelectionStart.Day.ToString();
}
```

Задание 4.

Создать форму авторизация. Приложение должно открываться формой для ввода пароля. Для создания формы можно использовать шаблон «Окно о программе». Доработать форму как представлено на рисунке



Если пароль введен верно, то открывается форма со вкладками, иначе выдается сообщение, что пароль введен не верно

Задание 5.

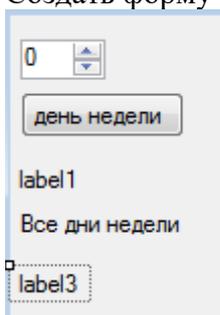
Доработать форму авторизации, таким образом, чтобы пользователю предоставлялось три попытки.

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

Создать новый проект

Задание 1.

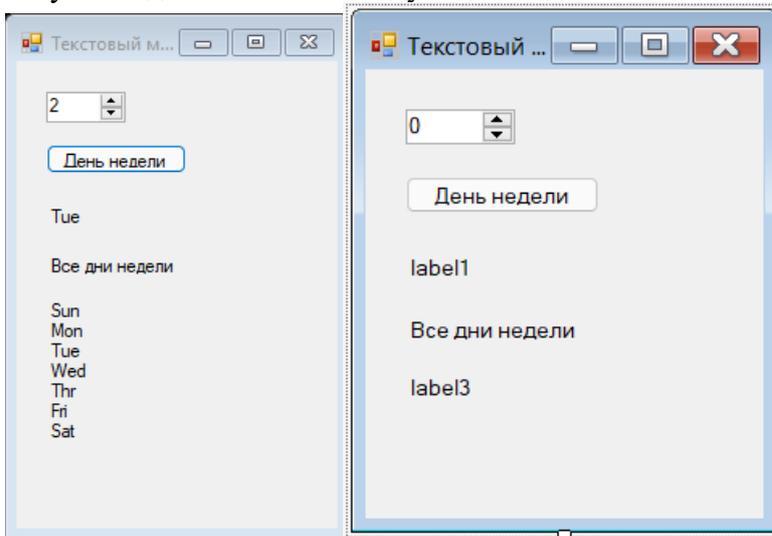
Создать форму как представлено на рисунке



Выбор номера дня недели используется в качестве значения индекса в текстовом массиве, содержащем название дней недели.

Создать процедуру обработки нажатия кнопки «день недели»

Результат должен иметь следующий вид



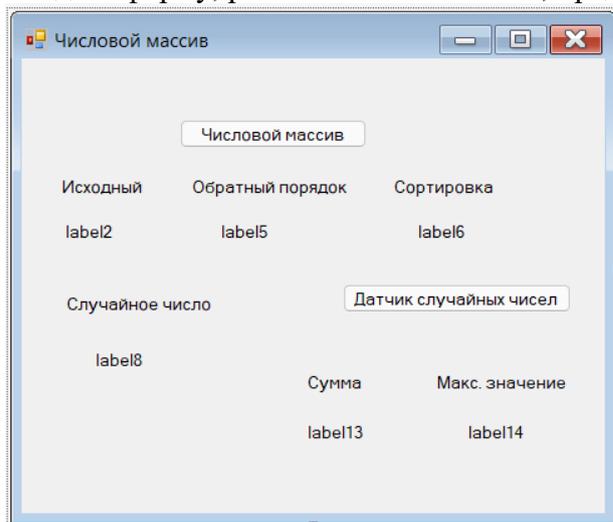
Предусмотреть чтобы при 0 label1 и label2 не отображались

Выполнить программу в пошаговом режиме.

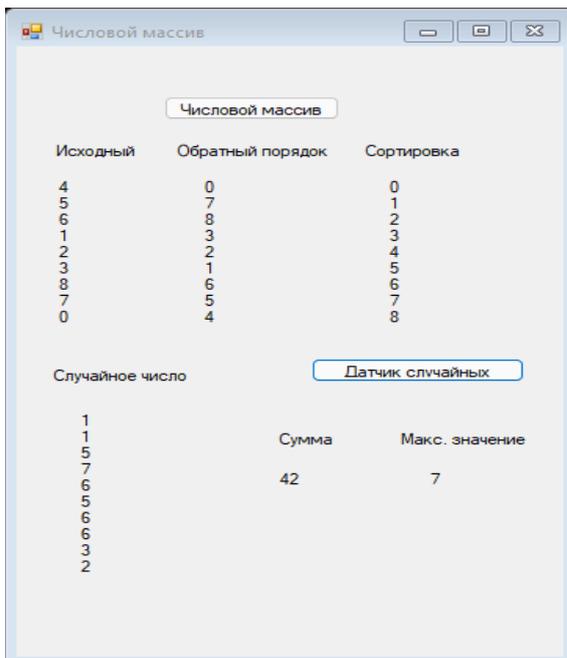
Задание 2.

Задан одномерный числовой массив

Создать форму, расположив на ней ЭУ, представленные на рисунке



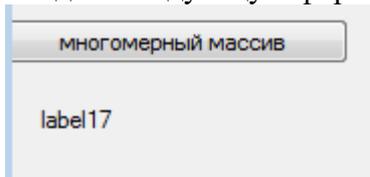
Создать процедуры отображения элементов массива, сортировки по возрастанию и для массива, заполненного случайно суммой элементов и максимальное значение



Установить точки останова, запустить программу на выполнение и контролировать значение переменных к окну проверки значений

Задание 3.

Создать следующую форму

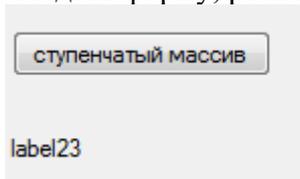


Разработать программу, которая выводит в метку многомерный массив (2-х) и максимальное значение в каждой строке

Выполнить программу в пошаговом режиме, осуществляя контроль вычисления максимума

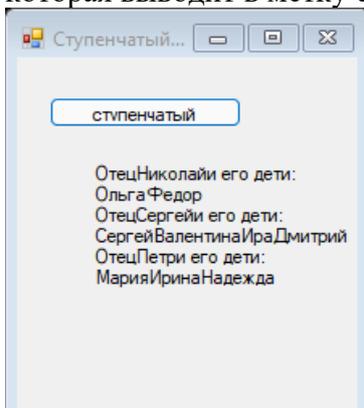
Задание 4.

Создать форму, разместив на ней кнопку и метку.



Исходные данные: имеется три родителя и их дети

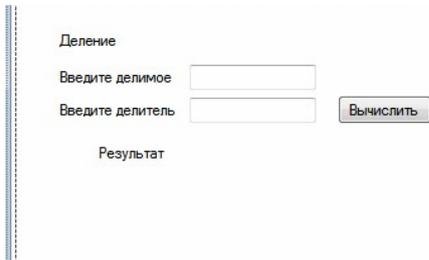
Требуется: Создать процедуру работы с массивом массивов (ломаным, ступенчатым), которая выводит в метку следующие данные



Установить точки останова, запустить программу на выполнение и контролировать значение переменных к окну проверки значений

Задание 5.

Создать обработчик исключений, который позволяет исключить ошибки времени выполнения



Программа должна анализировать, вводимые значения и вычислять результат деления числа 1 на число 2 и исключить не заполненные поля и значение делителя =0.

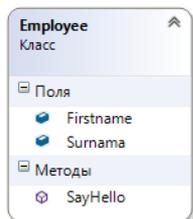
```
private void button2_Click(object sender, EventArgs e)
{
    try
    {
        double N, K, R;
        if (textBox5.Text == "")
        {
            textBox5.Select();
            throw new Exception("Поле \"Делимое\" не заполнено!");
        }
        if (textBox6.Text == "")
        {
            textBox6.Select();
            throw new Exception("Поле \"Делитель\" не заполнено!");
        }
        K = double.Parse(textBox5.Text);
        N = double.Parse(textBox6.Text);
        if (N == 0)
        {
            throw new Exception("Попытка деления на \"0\"");
        }
        R = K / N;
        label11.Text = R.ToString();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}
```

### Тема 3. Основы объектно-ориентированного программирования

#### Практическая работа 6. Создание классов

Задание 1.

Создать новый проект. Отрыть схему классов и создать класс Employee, представленный на рисунке



Задание 2.

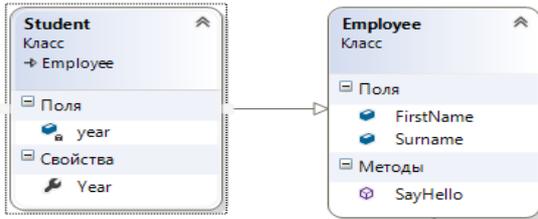
Доработать метод SayHello

```
public class Employee
{
    public string Surname;
    public string Firstname;

    public void SayHello(out string Fullname)
    {
        //throw new System.NotImplementedException();
        Fullname = "hello, " + Surname + " " + Firstname;
        return;
    }
}
```

Задание 3.

## Создать дочерний класс Student

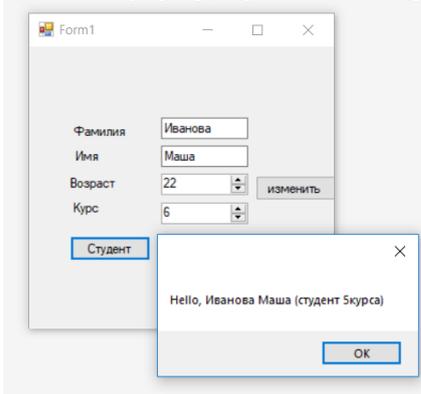


В классе Student есть закрытое поле курс, которое не может быть ниже единицы и больше пяти. Для управления доступом к этому полю используется свойство Year

```
public class Student : Employee
{
    private int year; //объявление закрытого поля
    //закрытое поле курс, которое не может быть ниже единицы и больше пяти
    public int Year; //объявление свойства
    {
        get // аксессор чтения поля
        {
            //throw new System.NotImplementedException();
            return year;
        }
        set // аксессор записи в поле
        {
            if (value < 1)
                year = 1;
            else if (value > 5)
                year = 5;
            else year = value;
        }
    }
}
```

### Задание 4.

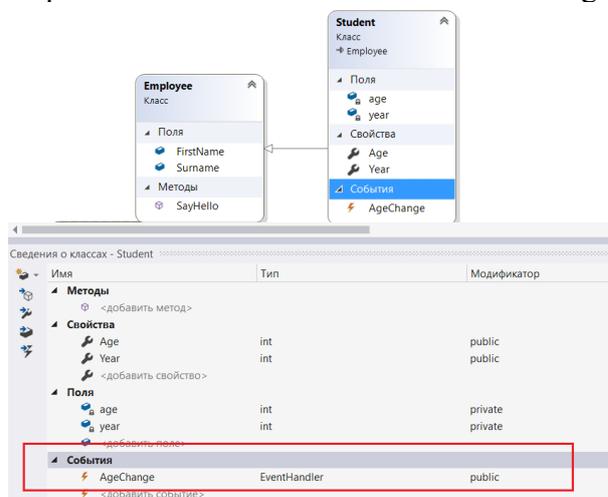
Создать форму, представленную на рисунке



Самостоятельно создать процедуру нажатия кнопки студент.

### Задание 5.

Перейти к схеме данных добавить поле Age, одноимённое свойство и метод

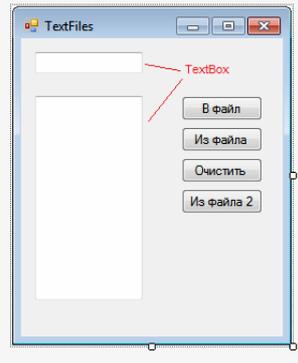


Доработать процедуру, установив ограничения на значение поля Age

## Практическая работа 7. Работа с файлами

### Задание 1.

Создать в новом проекте форму



### Задание 2.

Создать процедуру записи в файл

```
private void button1_Click(object sender, EventArgs e)
{
    StreamWriter myfile = null;
    try
    {
        myfile = new StreamWriter("C:/TEMP/textfile1.txt", true);
        myfile.WriteLine(textBox1.Text);
    }
    catch
    {
        MessageBox.Show("Ошибка записи в файл.");
        myfile.Close();
    }
    myfile.Close();
    textBox1.Clear();
}
```

### Задание 3.

Создать процедуру чтения из файла

```
private void button2_Click(object sender, EventArgs e)
{
    StreamReader myfile = null;
    string strbuf="";
    try
    {
        myfile = new StreamReader("C:/TEMP/textfile1.txt");
        textBox2.Clear();
        strbuf = myfile.ReadLine();
        while (strbuf != null)
        {
            textBox2.Text += strbuf+"\n";
            strbuf = myfile.ReadLine();
        }
    }
    catch
    {
        MessageBox.Show("Ошибка при чтении из файла.");
        myfile.Close();
        this.Close();
    }
    myfile.Close();
}
```

### Задание 4.

Самостоятельно создать процедуру чтения из файла используя метод ReadToEnd

### Задание 5.

Самостоятельно создать процедуру очистки содержимого файла

## Практическая работа 8. Создание MDI-приложений

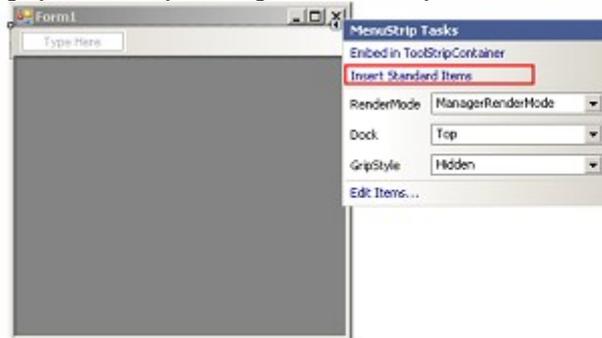
Создать в новом проекте MDI (Multiple Document Interface ) многодокументное приложение, позволяющее реализовать многооконный интерфейс в границах родительской формы

Задание 1.

Создать родительскую форму и изменить свойство **IsMDIContainer** в положение True.

Создать на форме стандартные меню и панели инструментов:

- Добавьте на форму ЭУ MenuStrip
- Выделите ЭУ MenuStrip1 и в правом верхнем углу выполнив щелчок по черному треугольнику выбрать команду Insert Standard Items(Вставить стандартные элементы)

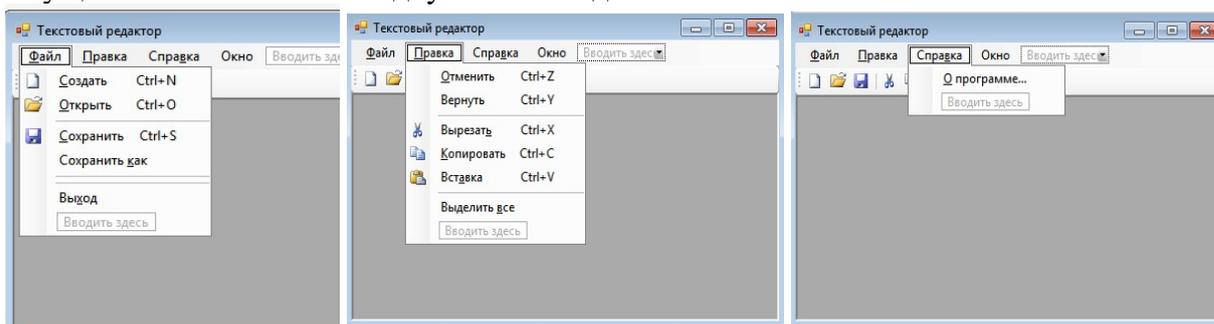


- Аналогичным образом добавить ЭУ ToolStrip и на нем разместить стандартные кнопки



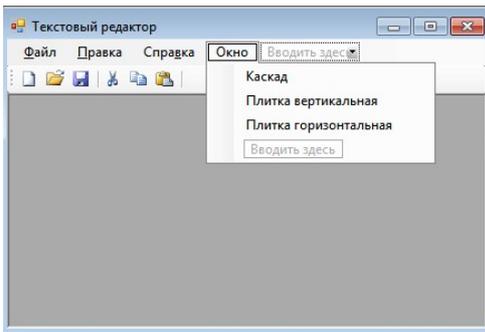
Задание 2.

Осуществить заполнение подпунктами созданное меню



Добавить пункт меню **Окно**, содержащее следующие подпункты:

- Каскад
- Плитка вертикальная
- Плитка горизонтальная



Задать следующие имена пунктам меню Окно

Пункт меню	Имя
Каскад	mnuCascade
Плитка вертикальная	mnuTileV
Плитка горизонтальная	mnuTileH

Добавить на родительскую форму следующие компоненты:

- Стандартный диалог открытия файлов - **OpenFileDialog**
- Стандартный диалог сохранения файлов - **SaveFileDialog**

Для обоих компонентов необходимо изменить свойство **Filter = файлы RTF**

Задание 3.

Создать дочернюю форму с именем ChildForm

Разместить на форме ЭУ RichTextBox, свойство Dock, которого установить Fill (Закрепить в родительском контейнере, т.е. выравнивание по всей клиентской области формы.)

Задание 4.

Кодирование дочерней формы

- Объявить две глобальные переменные

```
public partial class childForm : Form
{
    public bool textModified = false;
    public DialogResult retVal;
```

- Создать процедуру обработки события TextChanged

```
private void richTextBox1_TextChanged(object sender, EventArgs e)
{
    //документ был изменен пользователем
    textModified = true;
}
```

- Создать процедуру обработки события FormClosing

```
private void childForm_FormClosing(object sender, FormClosingEventArgs e)
{
    if (textModified)
    {
        retVal = MessageBox.Show("Сохранить изменения?", "Сохранение", MessageBoxButtons.YesNoCancel, MessageBoxIcon.Question);
        if ((retVal == DialogResult.Yes) || (retVal == DialogResult.Cancel))
        {
            e.Cancel = true;
        }
    }
}
```

Кодирование родительской формы

- Объявить переменную для работы с дочерней формы

```
public partial class MultyDocuementApp : Form
{
    private childForm cf;
```

- Создать процедуру обработки события FormClosing закрытия дочерней формы

```
private void cf_FormClosing(object sender, FormClosingEventArgs e)
{
    childForm AC = (childForm)this.ActiveMdiChild;
    if (e.Cancel)
    {
        if (AC.retVal == DialogResult.Yes)
        {
            сохранитькакToolStripMenuItem_Click(sender, e);
        }
    }
}
}
```

- Создать процедуры создания нового документа

//создание нового документа при помощи меню

```
private void создатьToolStripMenuItem_Click(object sender, EventArgs e)
{
    cf = new childForm();
    cf.MdiParent = this;
    cf.Text = "Новый документ";
    cf.Show();
    cf.FormClosing += cf_FormClosing;
}
}
```

//создание нового документа при помощи панели

```
private void создатьToolStripButton_Click(object sender, EventArgs e)
{
    cf = new childForm();
    cf.MdiParent = this;
    cf.Text = "Новый документ";
    cf.Show();
    cf.FormClosing += cf_FormClosing;
}
}
```

- Создать процедуру открытия существующего документа

//открытие документа при помощи меню

```
private void открытьToolStripMenuItem_Click(object sender, EventArgs e)
{
    if (openFileDialog1.ShowDialog() == DialogResult.OK)
    {
        cf = new childForm();
        cf.MdiParent = this;
        //необходимо в дизайнера дочерней формы поставить для ЭУ richTextBox модификатор доступа public
        cf.richTextBox1.LoadFile(openFileDialog1.FileName);
        cf.Text = openFileDialog1.FileName;
        cf.Show();
        cf.textModified = false;
        cf.FormClosing += cf_FormClosing;
    }
}
}
```

В обозревателе решения выберите файл дочерней формы и измените параметр доступа на public

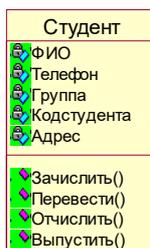
Задание 5.

Самостоятельно создать процедуры сохранения текущего документа и Сохранение текущего документа под новым именем

## Тесты

Тема 1.

1) Определить, что изображено на рисунке



- a) Состояние
  - b) Класс
  - c) Активность
  - d) Пакет
- 2) Указать, чем является «Группа»



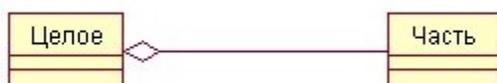
- a) Состояние
  - b) Класс
  - c) Атрибут
  - d) Операция
- 3) Определить представляют собой процессы, реализуемые некоторым классом
- a) Атрибуты
  - b) Операции
  - c) состояние
- 4) Указать, что является элементом информации, связанный с классом.
- a) Атрибут
  - b) Операция
  - c) состояние
- 5) Определить какой вид связи изображен на рисунке



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

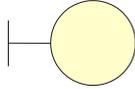


- a) Ассоциация
  - b) Обобщение
  - c) Агрегация
  - d) Композиция
- 7) Определить какой вид связи изображен на рисунке



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

- d) Композиция
- 8) Определить связь между целым и его частью.
- a) Ассоциация
- b) Обобщение
- c) Агрегация
- 9) Определить стереотип класса изображен на рисунке



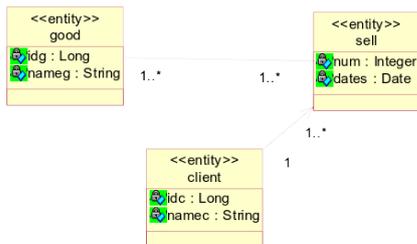
Страница регистрации

- a) Граничный класс
- b) Класс сущность
- c) Управляющий класс
- 10) Определить какой стереотип класса изображен на рисунке



Список клиентов

- a) Граничный класс
- b) Класс сущность
- c) Управляющий класс
- 11) Указать, что используется для моделирования статистической структуры объектов и связей между ними.
- a) диаграммы классов
- b) диаграммы активности
- c) диаграммы вариантов использования
- d) диаграмма последовательности
- 12) Определить какая диаграмма представлена на рисунке



- a) диаграммы классов
- b) диаграммы активности
- c) диаграммы вариантов использования
- d) диаграмма последовательности
- 13) Идентифицировать процессы, реализуемые некоторым классом
- a) Атрибут
- b) Операция
- c) Сообщение
- d) Имя
- 14) Определить свойство моделируемой сущности, которым обладают все объекты данного класса
- a) Атрибут
- b) Операция

- c) Сообщение
  - d) Имя
- 15) Идентифицировать хранение в одной структуре и данных, и методов
- a) свойством
  - a) полиморфизмом
  - b) наследованием
  - c) инкапсуляцией
- 16) Идентифицировать упорядочение абстракций, расположение их по уровням
- a) наследование
  - b) инкапсуляция
  - c) модульность
  - d) иерархия
- 17) Идентифицировать скрытие информации, т.е. маскировка всех внутренних деталей, не влияющих на внешнее поведение
- a) наследование
  - b) абстрагирование
  - c) полиморфизм
  - d) инкапсуляция
- 18) Идентифицировать 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) Определить 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) Определить связь между целым и его частью
- a) Агрегация
  - b) Ассоциация
  - c) Обобщение
  - d) Параметризация
- 21) Идентифицировать структурное отношение, показывающее, что объекты одного типа связаны с объектами другого типа
- a) Агрегация
  - b) Обобщение
  - c) Ассоциация
  - d) Композиция
- 22) Определить 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) Определить отношение, когда время жизни частей и целого совпадают и, когда объект-часть не может существовать без объекта-целого
- a) Агрегация

- b) Обобщение
- c) Ассоциация
- d) Композиция

24) Определить фрагмент какой диаграммы представлена на рисунке



- a) Прецедентов
- b) Активности
- c) Состояния

Ключи:

вопрос	1	2	3	4	5	6	7	8	9	10	11	12
ответ	b	b	b	a	a	b	c	c	a	b	a	a
вопрос	13	14	15	16	17	18	19	20	21	22	23	24
ответ	b	a	c	d	d	c	a	a	c	a	d	c

## Тема 2

- 1) Определить какое ключевое слово сообщает компилятору о необходимости определения типа переменной из выражения, находящегося с правой стороны оператора(C#)
  - a) float
  - b) char
  - c) variant
  - d) var
- 2) Определить правильное описание константы(C#)
  - a) float const Pi = 3.14;
  - b) char const Pi = 3.14;
  - c) const char Pi = 3.14;
  - d) const float Pi = 3.14;
- 3) Выбрать, что в каком формате будет выведен результат при помощи инструкции Label1.Text=String.Format("{0:C}",T)
  - a) даты
  - b) валюты
  - c) десятичного числа
  - d)
- 4) Выбрать правильное объявление строки символов.(C#)
  - a) float a;
  - b) string a;
  - c) char a;

- d) `int a;`
- 5) Выбрать правильное объявление числа с плавающей точкой.(C#)
- a) `float a;`
  - b) `string a;`
  - c) `char a;`
  - d) `int a;`
- 6) Выбрать правильное объявление целого числа.(C#)
- a) `float a;`
  - b) `string a;`
  - c) `char a;`
  - d) `int a;`
- 7) Указать проверку на равенство(C#)
- a) `a=b`
  - b) `a==b`
  - c) `a<>b`
  - d) `a!=b`
- 8) Указать проверку на неравенство(C#)
- a) `a=b`
  - b) `a==b`
  - c) `a<>b`
  - d) `a!=b`
- 9) Определить, 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) Определить, 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) Определить, 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) Определить, 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) Определить, что обозначает `a++`(C#)
- a) переменную a увеличить на 1
  - b) проверка на равенство 1
  - c) проверка на неравенство 1
- 14) Идентифицировать инструкцию, сообщающую программе о необходимости выполнения одного из множества условий (C#)
- a) `switch`
  - b) `select`
  - c) `if`
  - d) `do`
- 15) Выбрать 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) Указать какой цикл используется, когда известно количество итераций
- a) for
  - b) do
  - c) while
- 17) Указать какой цикл используется, когда известно количество итераций
- a) for
  - b) do
  - c) while
- 18) Указать какой цикл позволяет выполнять тело цикла до тех пор, пока указанное условие продолжает оставаться истинным
- a) for
  - b) do
  - c) while
- 19) Указать какой цикл удобно использовать при переборе элементов массива
- a) for
  - b) do
  - c) while
  - d) for each
- 20) Определить существуют ли VB массивы массивов (ступенчатый, jagged)
- a) да
  - b) нет
- 21) Определить существуют ли C# массивы массивов (ступенчатый, jagged)
- a) да
  - b) нет
- 22) Указать от какого базового класса C# массивы являются производными
- a) System.Array
  - b) Array
  - c) System.Data
  - d) Data.Array
- 23) Указать описание массива (C#).
- a) int[] a,b,c;
  - b) int() a,b,c;
  - c) int a(),b(),c();
  - d) int a[],b[],c[];
- 24) Указать описание ступенчатого или ломанного массива (C#).
- a) тип [][] имя массива = new тип[размер] [];
  - b) тип[,] имя = new тип [ разм\_1, разм\_2 ];
  - c) тип[] имя = new тип [ размерность ];
- 25) Выбрать описание двумерного массива (C#).
- a) тип [][] имя массива = new тип[размер] [];
  - b) тип[,] имя = new тип [ разм\_1, разм\_2 ];
  - c) тип[] имя = new тип [ размерность ];
- 26) Выбрать описание одномерного массива (C#).
- a) тип [][] имя массива = new тип[размер] [];
  - b) тип[,] имя = new тип [ разм\_1, разм\_2 ];
  - c) тип[] имя = new тип [ размерность ];
- 27) Указать какое окно удобно использовать для того чтобы выяснить, какие переменные находятся под влиянием текущего алгоритма

- a) Локальные
  - b) Интерпретация
  - c) Видимые
  - d) Контрольные значения
- 28) Указать в каком окне отображаются переменные только из текущей и предыдущей строк кода (C#)
- a) Локальные
  - b) Интерпретация
  - c) Видимые
  - d) Контрольные значения
- 29) Указать 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) Указать какое окно позволяет создавать индивидуальный список переменных
- a) Локальные
  - b) Интерпретация
  - c) ~Видимые
  - d) Контрольные значения
- 31) Указать в каком окне из окон, предназначенных для отладки программы можно выполнить следующие действия (рисунок1)
- a) Локальные
  - b) Интерпретация
  - c) Видимые
  - d) Контрольные значения
- 32) Выбрать , когда выполняется блок Finally, если в блоке Try возникает исключение
- a) До выполнения блока Catch
  - b) Вместо блока Catch
  - c) После выполнения блока Catch
  - d) Ни один вариант не подходит
- 33) Указать как правило оператор Try содержит \_\_\_\_ блок и хотя бы один блок \_\_\_\_
- a) Catch, Try
  - b) Try, Catch
  - c) Throw, Catch
  - d) Catch, Throw
- 34) Указать, что приложение игнорирует , если в блоке Try исключений не было.
- a) Блоки Catch
  - b) Оператор Return
  - c) Подходят оба предыдущих оператора
  - d) Ни один вариант не подходит
- 35) Выбрать какой оператор можно использовать оператор, чтобы сообщить о выполнении исключительных ситуаций в программе. (C#)
- a) Оператор Return
  - b) throw
  - c) exception
  - d) finally
  - e) error
- 36) Выбрать правильную инструкцию для обработки исключения на C#
- a) Catch Ex AS Exception

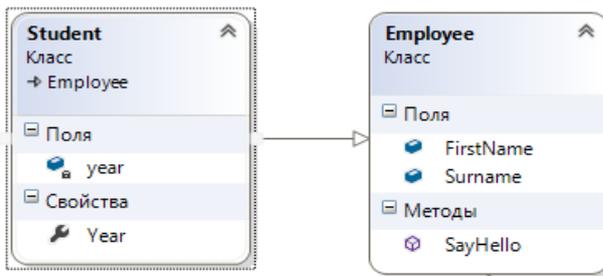
- b) catch (Exception ex)
  - c) Catch When Err.Number=13
- 37) Выбрать правильную инструкцию для обработки исключения на VB
- a) Catch Ex AS Exception
  - b) catch (Exception ex)
  - c) throw new Exception
- 38) Определить, что в Visual Studio позволяет добавлять элементы управления к форме с помощью визуальной технологии.
- a) Обзоратель решений
  - b) Стандартная
  - c) Панель Элементов
  - d) Обзоратель решений
- 39) Определить какое окно в Visual Studio позволяет просматривать файлы решения.
- a) Командный обзоратель
  - b) Обзоратель решений
  - c) Обзоратель объектов
  - d) Обзоратель серверов
- 40) Указать где в Visual Studio задается начальная форма (C#).
- a) в файле Program.vb
  - b) в файле Program.cs
  - c) В окне свойств объекта
  - d) В окне свойств проекта

КЛЮЧИ:

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

Тема 3.

- 1) Определить какие элементы в объектно-ориентированном представляют собой типы сходных объектов и в определенном смысле аналогичны строительным чертежам
  - a) Классы
  - b) Атрибуты
  - c) Свойства
  - d) Модели поведения
- 2) Указать свойство элемента управления CommandButton , позволяющее отобразить текст на кнопке
  - a) Value
  - b) Name
  - c) Text
  - d) Caption
- 3) Определить какой класс является дочерним



- a) Employer
  - b) Student
- 4) Указать какой вид доступа к классу возможен из любого места одной сборки либо из другой сборки, на которую есть ссылка
- a) public
  - b) internal
- 5) Указать какой элемент класса представляет собой переменную, объявленную внутри класса
- a) Поле
  - b) Свойство
  - c) Событие
  - d) Метод
- 6) Указать объекты какого этого класса могут обращаться к данному полю
- a) Private
  - b) Public
  - c) Protected
- 7) Указать к какому полю могут обращаться объекты любого класса
- a) Private
  - b) Public
  - c) Protected
- 8) Указать к какому полю могут обращаться только объекты классов-наследников
- a) Private
  - b) Public
  - c) Protected
- 9) Определить ,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) Определить, что представлено на рисунке
- ```

public string FirstName;
public string Surname;
  
```
- a) Свойство
  - b) Событие
  - c) Поле
  - d) Метод

- 11) Определить метод класса, предназначенный для инициализации объекта при его создании.
- Конструктор
  - Деструктор
- 12) Указать, какой принцип ООП позволяет создавать новый класс на базе другого.
- Наследование
  - Инкапсуляция
  - Полиморфизм
  - Абстрагирование
- 13) Определить, что представлено во фрагменте кода

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

- Класс Casher является дочерним от Employer
  - Класс Employer является дочерним от Casher
  - Создание экземпляра класса
- 14) Определить, что во фрагменте кода

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

- Класс Casher является дочерним от Employer
  - Класс Employer является дочерним от Casher
  - Создание экземпляра класса
- 15) Определить, что во фрагменте кода

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

- Объявление поля
  - Объявление свойства
  - Объявление метода
  - Объявление события
- 16) Указать что позволяет объектам реагировать на те или иные ситуации и выполнять необходимые ответные действия.
- Применение событий
  - Применение полей
  - Применение методов
  - Применение свойств
- 17) Указать какое утверждение верно
- Событие может вызываться только в том классе, где оно объявлено
  - Событие может вызываться любом классе, где оно объявлено
  - Событие может вызываться только в дочернем классе
- 

- 18) Определить, что представлено во фрагменте кода

```
public event EventHandler AgeChange;
```

- Объявление поля
  - Объявление свойства
  - Объявление метода
  - Объявление события
- 19) Указать, какой принцип ООП позволяет скрывать внутреннюю реализацию
- Инкапсуляция
  - Наследование

- c) Полиморфизм
  - d) Абстракция
- 20) Указать, какой принцип ООП определяет способность объектов с одним интерфейсом иметь различную реализацию.
- a) Инкапсуляция
  - b) Наследование
  - c) Полиморфизм
  - d) Абстракция
- 21) Указать, какой принцип ООП 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) Определить класс, позволяющий записывать строки в текстовый файл
- a) StreamWrite
  - b) StreamReader
- 23) Определить класс, позволяющий считывать строки из текстового файла
- a) StreamWrite
  - b) StreamReader
- 24) Определить метод класса, предназначенный для инициализации объекта при его создании
- a) Построитель
  - b) Деструктор
  - c) Конструктор
  - d) Инициализатор
- 25) Выбрать верное утверждение
- a) Событие может вызываться любом классе, где оно объявлено
  - b) Событие может вызываться только в дочернем классе
  - c) Событие может вызываться только в том классе, где оно объявлено
- 26) Определить метод, который имеет доступ к данным объекта, и его вызов выполняется через объект
- a) Статический
  - b) Простой
  - c) Конструктор
  - d) Деструктор
- 27) Указать внутри тела какого элемента класса используются специальные процедуры аксессоры (accessors) –Get и Set.
- a) события
  - b) метода
  - c) свойства
  - d) поля
- 28) Указать 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) Определить, 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) Определить, 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

Ключи:

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

### Вопросы для текущего контроля в форме устного опроса при защите практических работ по темам

Вопросы по теме 1.

1. Перечислить основные принципы ООА.
2. Дать определения элементов объектной модели.
3. Сделать обзор истории появления UML.
4. Определить место UML в процессе разработки ИС.
5. Определить использование UML на разных этапах разработки ПО.
6. Сделать обзор и объяснить назначение диаграмм UML.
7. Привести примеры диаграмм прецедентов
8. Привести примеры диаграмм активности
9. Привести примеры диаграмм состояния
10. Перечислить инструментальные средства объектно-ориентированного анализа.
11. Объяснить назначение функциональной модели системы.
12. Сделать обзор диаграммы классов.
13. Дать определение классу.
14. Рассмотреть в деталях виды связей классов

Вопросы по теме 2.

1. Дать определение понятию инструментальной среды разработки.
2. Сделать обзор инструментальной среды разработки Visual Studio.
3. Сделать обзор основных интерфейсов приложений Windows Forms.
4. Рассмотреть структуру программы VB.
5. Объяснить назначение проектов и приложений.
6. Объяснить назначение файлов и папок проектов.
7. Сделать обзор компонентов проекта.
8. Объяснить процесс компоновки и запуска приложения.
9. Рассмотреть в деталях настройку свойств формы.
10. Перечислить элементы управления для работы с текстом.
11. Перечислить элементы управления выбора.
12. Рассмотреть в деталях настройку свойств приложения.
13. Сделать обзор основных элементов управления.
14. Перечислить группирующие элементы управления.
15. Сделать обзор использования в форме графики.
16. Объяснить использование Полосы прокрутки.
17. Объяснить использование ЭУ таймер.
18. Объяснить использование строки состояния.

19. Описать процесс создание главного и контекстного меню.
20. Описать процесс создание панели элементов.
21. Перечислить типы ошибок.
22. Перечислить средства отладки приложений.
23. Объяснить назначение и способы использования контрольных точек.
24. Описать средства пошагового выполнения программы.
25. Сделать обзор инструментов для отладки программы.
26. Сделать обзор обработки исключительных ситуаций.
27. Сделать обзор структурной обработка исключений.

Вопросы по теме 3.

1. Дать определение принципам объектно-ориентированного программирования.
2. Рассмотреть в деталях способы создания классов.
3. Сделать обзор схемы классов.
4. Перечислить основные члены класса.
5. Дать определение закрытым, защищенным и открытым полям.
6. Определить статический и простой метод класса.
7. Описать создание класса наследника на основе базового класса.
8. Описать создание свойств класса.
9. Описать создание события класса.
10. Найти различие в кодирование полей и свойств.
11. Привести пример создание классов в C#.
12. Привести пример создание классов в VB.
13. Определить основные операции с файлами.
14. Перечислить классы для работы с текстовыми файлами
15. Сделать обзор класса StreamWriter
16. Описать процесс очистки файла
17. Сделать обзор класса с StreamReader.
18. Описать процесс считывания данных из текстового файла.
19. Описать процесс записи данных из текстового файла.
20. Рассмотреть в деталях открытие и создание файла для чтения и записи.
21. Перечислить основные интерфейсы приложений Windows Forms.
22. Сделать обзор однооконного интерфейса SDI.
23. Сделать обзор многооконного интерфейса MDI (Multiple Document Interface).
24. Привести примеры многооконного интерфейса.
25. Сделать обзор свойств главной формы.
26. Описать процесс отображения дочерней формы.
27. Сделать обзор расположения дочерних форм.
28. Сделать обзор использования стандартных диалоговых окон.
29. Привести примеры использования диалоговых окон для работы с файлами.
30. Перечислить способы работы с буфером обмена.
31. Привести примеры работы с буфером обмена.

## **2.2. Промежуточная аттестация**

### **Оценочные средства по дисциплине для промежуточной аттестации**

#### **2.2.1. Перечень вопросов для подготовки к дифференцированному зачету**

1. Сформулировать принципы объектно –ориентированного анализа.
2. Дать определение инструментальному средству, перечислить инструментальные средства объектно-ориентированного анализа и программирования.
3. Сделать обзор истории появления UML и определить место UML в процессе разработки ИС.
4. Дать определение классу, рассмотреть в деталях виды связей классов

5. Дать определение понятию инструментальной среды разработки, сделать обзор инструментальной среды разработки Visual Studio.
6. Дать определение проекту и решению, объяснить назначение файлов и папок проектов.
7. Сделать обзор основных интерфейсов приложений Windows Forms.
8. Объяснить процесс компоновки и запуска приложения.
9. Рассмотреть в деталях настройку свойств приложения VB.
10. Рассмотреть в деталях настройку свойств приложения C#.
11. Перечислить элементы управления для работы с текстом.
12. Перечислить элементы управления выбора.
13. Сделать обзор структуры программы на C#.
14. Сделать обзор простейших типов и выражений, привести примеры объявления переменных.
15. Сделать обзор конструкций ветвления (Конструкции IF ...Else, Switch) и привести примеры.
16. Сделать обзор циклических конструкций (For, For Each, While, Do).
17. Сделать обзор использования финансовых функций Excel в проектах C#, привести примеры.
18. Сделать обзор вызова функции VB в проектах C#, привести примеры.
19. Дать определение процессу отладки приложения, перечислить средства отладки и привести примеры.
20. Перечислить типы ошибок и сделать обзор обработки исключительных ситуаций.
21. Сформулировать принципы объектно-ориентированного программирования.
22. Дать определение классу, рассмотреть в деталях способы создания класса, сделать обзор окна схемы классов.
23. Дать определение классу, перечислить члены класса.
24. Дать определение полю, уточнить назначение закрытых, защищенных и открытых полей.
25. Дать определение методу, определить статический и простой методы класса.
26. Дать определение свойства класса, описать создание свойств, привести примеры.
27. Дать определение события класса, описать создание событий, привести примеры.
28. Описать создание класса наследника на основе базового класса.
29. Определить основные операции с файлами, перечислить классы для работы с текстовыми файлами.
30. Сделать обзор однооконного интерфейса SDI и многооконного интерфейса MDI , перечислить примеры.
31. Дать определение главной формы интерфейса MDI, сделать обзор свойств главной формы.
32. Дать определение дочерней формы интерфейса MDI, описать процесс отображения дочерних форм.

### 3. Описание системы оценивания, шкала оценивания

#### 3.1. Показатели и критерии оценивания для текущего контроля.

**Опрос (О)** - это основной вид устной или письменной проверки, может использоваться как фронтальный (краткие ответы, как правило, с места на вопросы преподавателя по сравнительно небольшому объему материала), так и индивидуальный (проверка знаний отдельных обучающихся). Комбинированный опрос - одновременный вызов для ответа сразу нескольких обучающихся, из которых один отвечает устно, один-два готовятся к ответу, выполняя на доске различные записи, а остальные выполняют за отдельными столами индивидуальные письменные или практические задания преподавателя.

Критерии оценивания:

*Оценки «отлично»* заслуживает студент, правильно ответивший на вопрос;

*Оценки «хорошо»* заслуживает студент, в целом правильно ответивший на вопрос, но допустивший незначительные ошибки и неточности;

*Оценки «удовлетворительно»* заслуживает студент, 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.

**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.

**Контрольная работа (КР)** - письменная работа по теме. Состоит из нескольких задач различной степени сложности.

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.

### 3.2. Показатели и критерии оценивания для промежуточного контроля

ПК 5.1 Собирать исходные данные для разработки проектной документации на информационную систему.	Оценка <b>«отлично»</b> - сформулирована задача по обработке информации; выполнен анализ предметной области; выполнены сбор и обработка исходной информации с помощью инструментальных средств. <b>Дополнительно для квалификаций " Специалист по информационным системам" и "Разработчик web и</b>	Дифференцированный зачет
--	--	--------------------------

	<p><b>мультимедийных приложений":</b>  Построена и обоснована модель информационной системы; выбраны и обоснованы средства реализации информационной системы.  Оценка «<b>хорошо</b>» - сформулирована задача по обработке информации; выполнен анализ предметной области; собрана исходная информация; выполнена обработка исходной информации с помощью инструментальных средств.  <b>Дополнительно для квалификаций " Специалист по информационным системам" и "Разработчик web и мультимедийных приложений":</b>  Построена и обоснована модель информационной системы; выбраны и обоснованы средства реализации информационной системы.  Оценка «<b>удовлетворительно</b>» - сформулирована задача по обработке информации; выполнен анализ предметной области; собрана исходная информация; частично выполнена обработка исходной информации с помощью инструментальных средств.  <b>Дополнительно для квалификаций " Специалист по информационным системам" и "Разработчик web и мультимедийных приложений":</b>  Построена модель информационной системы; выбраны средства реализации информационной системы.</p>	
<p>ПК 5.2 Разрабатывать проектную документацию на разработку информационной системы в соответствии с требованиями заказчика.</p>	<p>Оценка «<b>отлично</b>» - требования клиента проанализированы, предложен и обоснован математический алгоритм решения задачи по обработке информации; указаны стандарты на оформление алгоритмов; предложенный алгоритм оформлен в соответствии с требованиями стандартов.  Оценка «<b>хорошо</b>» - требования клиента проанализированы, предложен математический алгоритм решения задачи по обработке информации; предложенный алгоритм оформлен в соответствии с требованиями стандартов.  Оценка «<b>удовлетворительно</b>» - требования клиента проанализированы, предложен математический алгоритм решения задачи по обработке информации; предложенный алгоритм оформлен в соответствии с требованиями стандартов с некоторыми отклонениями.</p>	
<p>ПК 5.3 Разрабатывать подсистемы безопасности информационной</p>	<p>Оценка «<b>отлично</b>» - разработан проект подсистемы безопасности информационной системы, в спецификации отражены задачи проекта в полном объеме.</p>	

<p>системы в соответствии с техническим заданием.</p>	<p><b>Дополнительно для квалификаций " Специалист по информационным системам" и "Разработчик web и мультимедийных приложений":</b>  В проекте предусмотрен файловый ввод-вывод; разработаны клиентская и серверная часть проекта; при разработке использованы языки структурного, объектно-ориентированного программирования и языка сценариев; разработан графический интерфейс приложения в соответствии с принципами проектирования GUI.  Оценка «<b>хорошо</b>» - разработан проект подсистемы безопасности информационной системы, в спецификации отражены основные задачи проекта.</p> <p><b>Дополнительно для квалификаций " Специалист по информационным системам" и "Разработчик web и мультимедийных приложений":</b>  В проекте предусмотрен файловый ввод-вывод; разработаны основные функции клиентской и серверной части проекта; при разработке использованы языки структурного, объектно-ориентированного программирования и языка сценариев; разработан графический интерфейс приложения в соответствии с принципами проектирования GUI.  Оценка «<b>удовлетворительно</b>» - разработан проект подсистемы безопасности информационной системы, в спецификации отражены задачи проекта с некоторыми недочетами.</p> <p><b>Дополнительно для квалификаций " Специалист по информационным системам" и "Разработчик web и мультимедийных приложений":</b>  В проекте частично реализован файловый ввод-вывод; разработаны основные функции клиентской и серверной части проекта; при разработке использованы языки структурного, объектно-ориентированного программирования и языка сценариев; частично разработан графический интерфейс приложения.</p>	
<p>ПК 5.4 Производить разработку модулей информационной системы в соответствии с техническим заданием.</p>	<p>Оценка «<b>отлично</b>» - разработаны варианты возможных решений, выбран и обоснован оптимальный на основе анализа интересов клиента; разработаны модули информационной системы; при разработке использованы языки структурного, объектно-ориентированного программирования и языка сценариев; разработана документация на модули (по перечню в задании); выполнена оценка</p>	

	<p>качества разработанных модулей по выбранным и обоснованным метрикам.</p> <p><b>Дополнительно для квалификаций " Специалист по информационным системам" и "Разработчик web и мультимедийных приложений":</b></p> <p>Разработан проект, в проекте разработан графический интерфейс приложения в соответствии с принципами проектирования GUI.</p> <p>Оценка «<b>хорошо</b>» - разработан и обоснован вариант возможного решения, на основе анализа интересов клиента; разработаны модули информационной системы; при разработке использованы языки структурного, объектно-ориентированного программирования и языка сценариев; разработана документация на модули (по перечню в задании); выполнена оценка качества разработанных модулей по набору метрик.</p> <p><b>Дополнительно для квалификаций " Специалист по информационным системам" и "Разработчик web и мультимедийных приложений":</b></p> <p>Разработан проект, в проекте разработан графический интерфейс приложения в соответствии с принципами проектирования GUI.</p> <p>Оценка «<b>удовлетворительно</b>» - разработан вариант возможного решения; разработаны модули информационной системы; при разработке использованы языки структурного, объектно-ориентированного программирования и языка сценариев; разработана документация на модули (по перечню в задании); выполнена оценка качества разработанных модулей по набору метрик.</p> <p><b>Дополнительно для квалификаций " Специалист по информационным системам" и "Разработчик web и мультимедийных приложений":</b></p> <p>Разработан проект, в проекте разработан графический интерфейс приложения.</p>	
--	---	--