**МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ ПМР**

**ГОУ СПО «ТИРАСПОЛЬСКИЙ ТЕХНИКУМ ИНФОРМАТИКИ И ПРАВА»**

**ДНЕВНИК**

**ПРОХОЖДЕНИЯ практики**

**\_\_УП.03 учебная практика**

вид практики

Профессиональный модуль \_\_ПМ.03 Участие в интеграции программных модулей\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

индекс, наименование профессионального модуля

Междисциплинарный курс \_МДК.03.01 Технология разработки программного обеспечения

\_МДК.03.02 Инструментальные средства разработки программного обеспечения\_\_\_\_\_\_\_\_\_

\_МДК.03.03 Документирование и сертификация\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

индекс, наименование междисциплинарного курса

Ф.И.О. студента\_\_\_\_\_\_\_\_\_**Неделько Никита Михайлович**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Специальность \_2.09.02.03 Программирование в компьютерных системах \_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

код, наименование специальности

Курс \_\_\_\_\_\_\_\_\_\_\_\_\_\_4\_\_\_\_\_\_\_\_\_\_\_\_\_\_ группа \_\_\_\_\_\_\_\_\_\_\_\_413\_\_\_\_\_\_\_\_\_\_

Время прохождения практики: с «27» февраля 2023г. по «11» марта 2023г.

База практики \_\_\_\_ГОУ СПО «Тираспольский техникум информатики и права»\_\_\_\_\_\_\_\_\_

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№**  **п/п** | **Дата** | **Описание выполненной работы** | **Количество часов** | **Оценка и подпись руководителя практики** |
|  | 27.02.2023 | Вводная беседа по теме практики. Цели и задачи практики. Вводный инструктаж по технике безопасности во время прохождения практики | 2 |  |
|  | 27.02.2023 | Обзор современных основных инструментальных средств разработки программных продуктов. Разработка алгоритма и спецификаций структурных компонентов | 2 |  |
|  | 27.02.2023 | Обзор современных основных инструментальных средств разработки программных продуктов | 2 |  |
|  | 28.02.2023 | Основные этапы разработки программного продукта | 2 |  |
|  | 28.02.2023 | Проектирование программного обеспечения на уровне модулей. Изучение алгоритма взаимодействия форм в многодокументном приложении | 2 |  |
|  | 28.02.2023 | Диаграммы переходов состояний | 2 |  |
|  | 01.03.2023 | Функциональные диаграммы | 2 |  |
|  | 01.03.2023 | Диаграммы потоков данных | 2 |  |
|  | 01.03.2023 | Диаграмма «сущность—связь» | 2 |  |
|  | 02.03.2023 | Диаграмма UMA | 2 |  |
|  | 02.03.2023 | Структурная схема. Структурный анализ | 2 |  |
|  | 02.03.2023 | Методы анализа, ориентированные на структуры данных. Метод анализа Джексона | 2 |  |
|  | 03.03.2023 | Функциональная схема | 2 |  |
|  | 03.03.2023 | Способ тестирования базового пути. Способы тестирования условий. | 2 |  |
|  | 03.03.2023 | Способ тестирования потоков данных. Тестирование циклов | 2 |  |
|  | 04.03.2023 | Диаграммы вариантов использования | 2 |  |
|  | 04.03.2023 | Диаграммы деятельности | 2 |  |
|  | 04.03.2023 | Диаграммы последовательности | 2 |  |
|  | 06.03.2023 | Диаграмма классов | 2 |  |
|  | 06.03.2023 | Структурирование системы. Моделирование управления | 2 |  |
|  | 06.03.2023 | Создание, отладка и тестирование инструментального программного обеспечения | 2 |  |
|  | 07.03.2023 | Разработка концепции современной интегрированной среды разработки приложений | 2 |  |
|  | 07.03.2023 | Технология компонентной разработки COM | 2 |  |
|  | 07.03.2023 | Технология Java | 2 |  |
|  | 08.03.2023 | Платформа .NET Framework | 2 |  |
|  | 08.03.2023 | Общие сведения о CASE-средствах | 2 |  |
|  | 08.03.2023 | Принципы построения и приемы работы с CASE-средствами | 2 |  |
|  | 09.03.2023 | Основные функциональные возможности CASE-средств | 2 |  |
|  | 09.03.2023 | Классификация CASE-средств | 2 |  |
|  | 09.03.2023 | Обзор современных CASE-средств | 2 |  |
|  | 10.03.2023 | Спецификация требований программного обеспечения | 2 |  |
|  | 10.03.2023 | Управление документированием программного обеспечения | 2 |  |
|  | 10.03.2023 | Оформление пакета документов по прохождение раздела учебной практики | 2 |  |
|  | 11.03.2023 | Сборка и отладка программы в полном объёме, подготовка презентаций для защиты | 2 |  |
|  | 11.03.2023 | Разработка пояснительной записки в соответствии с техническим заданием. | 2 |  |
|  | 11.03.2023 | Дифференцированный зачет | 2 |  |

Содержание объемов выполненных работ подтверждаю

Руководитель практики от техникума (предприятия) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/

# 27.02.2023

**Тема:** Основные этапы разработки программного продукта. Проектирование программного обеспечения на уровне модулей. Изучение алгоритма взаимодействия форм в многодокументном приложении. Диаграммы переходов состояний.

**Цель:** Изучить тему:Основные этапы разработки программного продукта. Проектирование программного обеспечения на уровне модулей. Изучение алгоритма взаимодействия форм в многодокументном приложении. Диаграммы переходов состояний.

## Ход работы

### Заменить отрицательные элементы массива положительными элементами.

#### Код:

Start:

try

{

InputSize:

Console.Write("Введите размер массива: ");

string? sizeStr = Console.ReadLine();

if (sizeStr == null)

{

Console.WriteLine("Размер массива обязателен");

goto InputSize;

}

int size = int.Parse(sizeStr);

int[] arr = new int[size];

Console.WriteLine("Введите элементы массива:");

for (int i = 0; i < arr.Length; i++)

{

InputEl:

string? elStr = Console.ReadLine();

if (elStr == null)

{

Console.WriteLine("Элемент массива обязателен");

goto InputEl;

}

arr[i] = int.Parse(elStr);

}

for (int i = 0; i < arr.Length; i++)

{

if (arr[i] < 0)

{

arr[i] = -arr[i];

}

}

Console.WriteLine("Массив с положительными элементами:");

for (int i = 0; i < arr.Length; i++)

{

Console.Write(arr[i] + " ");

}

}

catch (Exception ex)

{

Console.WriteLine(ex.ToString());

goto Start;

}

#### Консоль:

|  |
| --- |
| Рис. 1. Вид консоли |

### Дан массив целых чисел из n элементов, заполненный случайным образом числами из промежутка [-10,10]. Найти сумму четных отрицательных элементов. Вывести индексы тех элементов, значения которых кратны 3 и 7. Определить, есть ли пара соседних элементов с суммой, равной заданному числу.

#### Код:

using System;

class Program

{

static void Main(string[] args)

{

int[] arr = new int[10];

Random rand = new Random();

Console.WriteLine("Исходный массив:");

for (int i = 0; i < arr.Length; i++)

{

arr[i] = rand.Next(-10, 11);

Console.Write(arr[i] + " ");

}

int sum = 0;

int index = -1;

Console.WriteLine("\nИндексы кратных 3 и 7:");

for (int i = 0; i < arr.Length; i++)

{

if (arr[i] < 0 && arr[i] % 2 == 0)

{

sum += arr[i];

}

if (arr[i] % 21 == 0)

{

Console.Write(i + " ");

}

}

InputN:

Console.WriteLine("\nОпределить, есть ли пара соседних элементов с суммой, равной заданному числу: ");

string? nStr = Console.ReadLine();

if (nStr == null)

{

Console.WriteLine("Число обязателен");

goto InputN;

}

int n = int.Parse(nStr);

for (int i = 0; i < arr.Length; i++)

{

if (i > 0 && arr[i - 1] + arr[i] == n)

{

index = i - 1;

break;

}

}

Console.WriteLine("\nСумма четных отрицательных элементов:");

if (sum == 0)

{

Console.WriteLine("Нет четных отрицательных элементов.");

}

else

{

Console.WriteLine(sum);

}

if (index == -1)

{

Console.WriteLine("Нет соседних элементов с суммой, равной " + n + ".");

}

else

{

Console.WriteLine("Соседние элементы с суммой, равной " + n + ", находятся на позициях " + index + " и " + (index + 1) + ".");

}

}

}

#### Консоль:

|  |
| --- |
| Рис. 2. Вид консоли |

### Сформировать и вывести следующую матрицу:

#### Код:

int countRows = 5;

int countCols = 5;

int[,] matrix = new int[countRows, countCols];

for (int i = 0; i < countRows; i++)

{

for (int j = 0; j < countCols; j++)

{

matrix[i, j] = countRows - i;

}

}

for (int i = 0; i < countRows; i++)

{

for (int j = 0; j < countCols; j++)

{

Console.Write(matrix[i, j] + " ");

}

Console.WriteLine();

}

#### Консоль:

|  |
| --- |
| Рис. 3. Вид консоли |

**Вывод:** Изучил тему:Основные этапы разработки программного продукта. Проектирование программного обеспечения на уровне модулей. Изучение алгоритма взаимодействия форм в многодокументном приложении. Диаграммы переходов состояний.

# 28.02.2023

**Тема:** Вводная беседа по теме практики. Цели и задачи практики. Вводный инструктаж по технике безопасности во время прохождения практики. Обзор современных основных инструментальных средств разработки программных продуктов. Разработка алгоритма и спецификаций структурных компонентов. Обзор современных основных инструментальных средств разработки программных продуктов.

**Цель:** Изучить тему:Вводная беседа по теме практики. Цели и задачи практики. Вводный инструктаж по технике безопасности во время прохождения практики. Обзор современных основных инструментальных средств разработки программных продуктов. Разработка алгоритма и спецификаций структурных компонентов. Обзор современных основных инструментальных средств разработки программных продуктов.

## Ход работы

### Создать форму расчета времени и стоимости поездки по городу. Вводится расстояние в километрах и выбирается вид транспорта. При поездке на автомобиле средняя скорость составляет 40 км/ч, расход бензина 10 л/ 100 км, стоимость бензина 47 руб./литр. При поездке на метро средняя скорость составляет 40 км/ч, а стоимость проезда 60 руб. При поездке на мотоцикле средняя скорость составляет 60 км/ч, расход бензина 5 л/100 км, стоимость бензина такая же, как и для автомобиля. Рассчитать, сколько времени потребуется, чтобы добраться до места назначения, и стоимость поездки.

#### Код:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace \_28\_02\_2023\_1

{

public partial class Form1 : System.Windows.Forms.Form

{

public Form1()

{

InitializeComponent();

}

private void buttonCalc\_Click(object sender, EventArgs e)

{

if (comboBoxTransport.SelectedIndex == -1) return;

labelResultTime.Text = $"{getTime()} ч.";

labelResultPrice.Text = $"{getPrice()} руб.";

}

private int getSpeed()

{

int speed = 0;

switch (comboBoxTransport.SelectedIndex)

{

case 0:

speed = 40;

break;

case 1:

speed = 40;

break;

case 2:

speed = 60;

break;

}

return speed;

}

private decimal getPrice()

{

decimal distance = numericUpDownDistance.Value;

decimal fullPrice = 0;

decimal kmPrice = 0;

switch (comboBoxTransport.SelectedIndex)

{

case 0:

kmPrice = decimal.Parse(((10 / (100 \* 1.0)) \* 47).ToString());

break;

case 1:

fullPrice = 60;

break;

case 2:

kmPrice = decimal.Parse(((5 / (100 \* 1.0)) \* 47).ToString());

break;

}

if (fullPrice == 0) fullPrice = kmPrice \* distance;

return fullPrice;

}

private decimal getTime()

{

decimal distance = numericUpDownDistance.Value;

int speed = getSpeed();

return distance / speed;

}

}

}

#### Форма:

|  |
| --- |
| Рис. 4. Вид формы |

**Вывод:** Изучил тему:Вводная беседа по теме практики. Цели и задачи практики. Вводный инструктаж по технике безопасности во время прохождения практики. Обзор современных основных инструментальных средств разработки программных продуктов. Разработка алгоритма и спецификаций структурных компонентов. Обзор современных основных инструментальных средств разработки программных продуктов.

# 01.03.2023

**Тема:** Функциональные диаграммы. Диаграммы потоков данных. Диаграмма «сущность—связь».

**Цель:** Изучить тему:Функциональные диаграммы. Диаграммы потоков данных. Диаграмма «сущность—связь».

## Ход работы

Исследовать информационную область:

1. Анализ полученных знаний для последующего моделирования;
2. Разработка модели процесса в стандарте IDEF0;
3. Описания процессов в стандарте IDEF3;
4. Разработка смешанной модели описания процесса на основе стандартов IDEF0 и IDEF3.
5. Создание сценариев работы системы;
6. Создание математической модели данного объекта исследования.
7. Сравнительный анализ.
8. Применение методов обследования предприятия;
9. Описание документооборота и обработки информации в стандарте DFD;
10. Разработка смешанной модели описания процесса на основе стандартов IDEFO, DFD и IDEF3.
11. Построение структурной схемы предметной области;
12. Создание математической модели данного объекта исследования.
13. Сравнительный анализ.

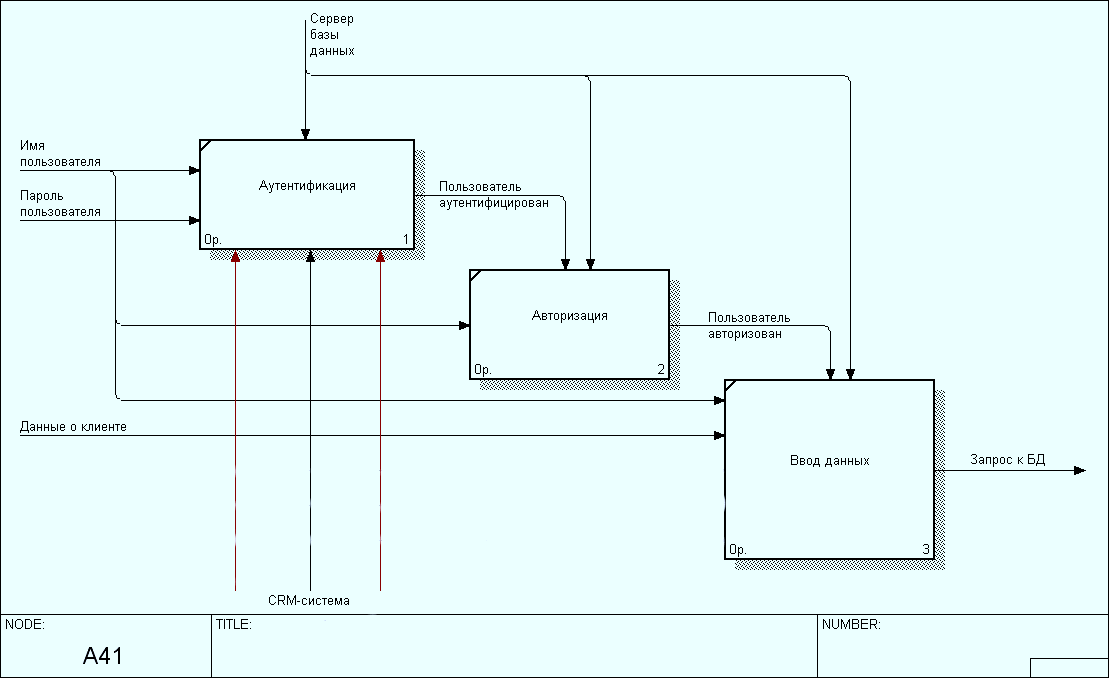


Рис. 5. Модель процесса IDEF0

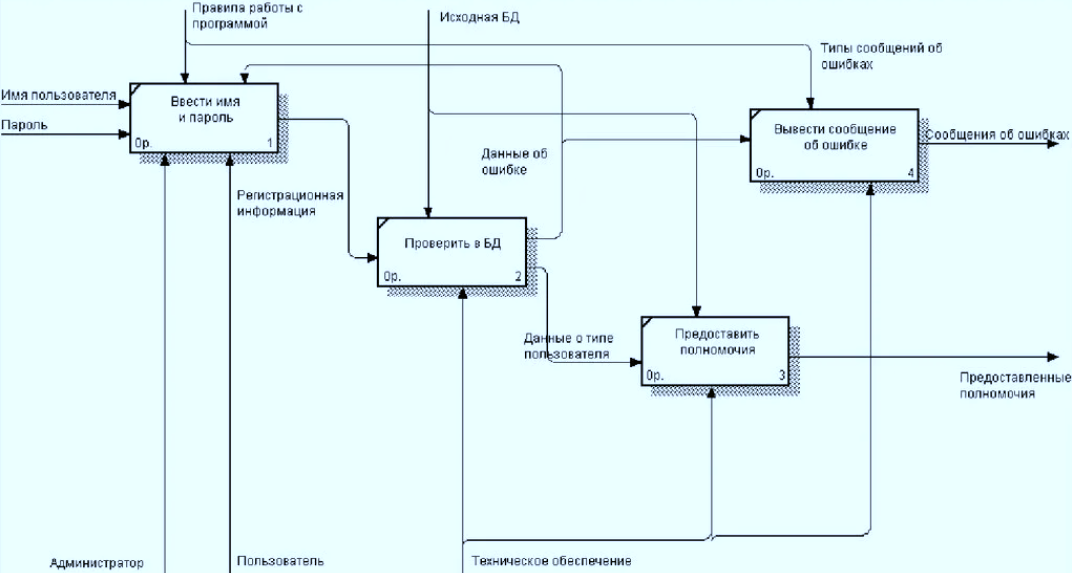


Рис. 6. Диаграмма потоков данных

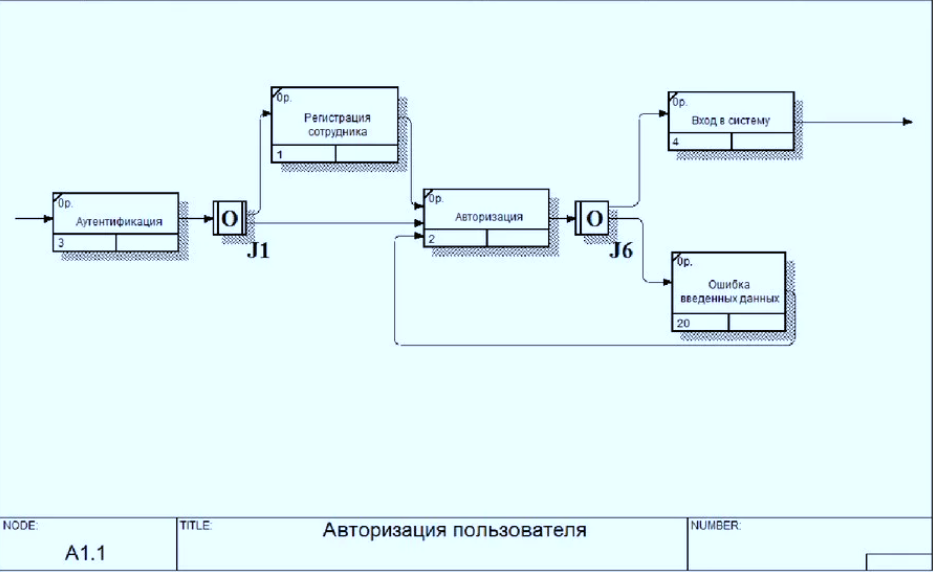


Рис. 7. Диаграмма выбора IDEF3

**Вывод:** Изучил тему:Функциональные диаграммы. Диаграммы потоков данных. Диаграмма «сущность—связь».

# 03.03.2023

**Тема:** Функциональная схема. Способ тестирования потоков данных. Тестирование циклов. Способ тестирования базового пути. Способы тестирования условий.

**Цель:** Изучить тему:Функциональная схема. Способ тестирования потоков данных. Тестирование циклов. Способ тестирования базового пути. Способы тестирования условий.

## Ход работы

### Разработать приложение, выполняющее построение графика функции y = cos 2x + cosx в заданных пределах.

#### Код:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace \_02\_03\_2023\_1

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void buttonChart\_Click(object sender, EventArgs e)

{

double from = double.Parse(numericUpDownFrom.Value.ToString());

double to = double.Parse(numericUpDownTo.Value.ToString());

double step = 0.1;

chart.Series[0].Points.Clear();

double x = from;

double y = 0;

while (x <= to)

{

y = Math.Cos(2 \* x) + Math.Cos(x);

chart.Series[0].Points.AddXY(x, y);

x += step;

}

}

}

}

#### Форма:

|  |
| --- |
| Рис. 8. Вид формы |

### Разработать приложение, которое используя графические примитивы, выводит флаг и герб нашей республики.

#### Код:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace \_02\_03\_2023\_2

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

Graphics g;

private void Form1\_Paint(object sender, PaintEventArgs e)

{

g = CreateGraphics();

g.Clear(Color.Red);

Brush brush = new SolidBrush(Color.Yellow);

g.FillRectangle(brush, 180, 100, 40, 200);

Point[] molot = { new Point(100, 100), new Point(300, 100), new Point(300, 60), new Point(140, 60) };

g.FillPolygon(brush, molot);

System.Drawing.Drawing2D.GraphicsPath path = new System.Drawing.Drawing2D.GraphicsPath();

path.AddArc(100, 40, 200, 200, 180, -135);

Pen pen = new Pen(Color.Yellow, 40);

g.DrawPath(pen, path);

Point[] ruchka = { new Point(260, 200), new Point(300, 200), new Point(340, 260), new Point(300, 260) };

g.FillPolygon(brush, ruchka);

}

}

}

#### Форма:

|  |
| --- |
| Рис. 9. Вид формы |

**Вывод:** Изучил тему:Функциональная схема. Способ тестирования потоков данных. Тестирование циклов. Способ тестирования базового пути. Способы тестирования условий.

# 04.03.2023

**Тема:** Диаграммы вариантов использования. Диаграммы деятельности. Диаграммы последовательности.

**Цель:** Изучить тему:Диаграммы вариантов использования. Диаграммы деятельности. Диаграммы последовательности.

## Ход работы

### Построить диаграммы вариантов использования, диаграмму состояний, диаграмму деятельности и диаграммы последовательности в соответствии с вариантом.

|  |
| --- |
| Рис. 10. Диаграмма вариантов использования |
| Рис. 11. Диаграмма состояний |
| Рис. 12. Диаграмма деятельности |
| Рис. 13. Диаграмма последовательности |

**Вывод:** Изучил тему:Диаграммы вариантов использования. Диаграммы деятельности. Диаграммы последовательности.

# 06.03.2023

**Тема:** Диаграмма классов. Структурирование системы. Моделирование управления. Создание, отладка и тестирование инструментального программного обеспечения.

**Цель:** Изучить тему:Диаграмма классов. Структурирование системы. Моделирование управления. Создание, отладка и тестирование инструментального программного обеспечения.

## Ход работы

### Построить диаграмму классов в соответствии с вариантом.

|  |
| --- |
| Рис. 14. Диаграмма классов интернет-магазина |

### Ответить на вопросы:

**Назовите основные элементы диаграммы классов?**

Диаграмма классов содержит: названия классов, свойства классов, типы свойств классов и связи между классами.

**Какие виды связей доступны в диаграмме классов?**

В диаграмме классов используются следующие виды связей: Unidirectional association (однонаправленная ассоциация); Dependency (зависимость); Association class (ассоциированный класс); Generalization (наследование); Realization (реализация).

**Для чего используется каждый вид связи?**

Unidirectional association– это один из важных и сложных типов связи. Она показывает, что один класс включается в другой как атрибут по ссылке или по значению.

Тип связи Dependency позволяет показать, что один класс использует объекты другого. Это может осуществляться при передаче параметров или вызове операций класса. В таком случае генератор кода Rational Rose включает заголовочный файл в класс, который использует операторы или объекты другого класса.

Тип связи Association class используется для отображения свойства ассоциации. Свойства сохраняются в классе и соединяются связью Association. тип не имеет своих спецификаций. Ассоциация предназначена для задания дополнительных атрибутов у связи. Она обозначает, что некоторый класс со своими атрибутами включается как элемент в два других, хотя при генерации кода это не отображается.

Тип связи Generalization позволяет указать, что один класс является родительским по отношению к другому, при этом будет создан код наследования класса.

Тип связи Realization позволяет показать, что один класс является реализацией, т.е. создан на основе шаблона другого. В Rational Rose для обозначения класса шаблона используется понятие параметризованный класс.

**Вывод:** Изучил тему:Диаграмма классов. Структурирование системы. Моделирование управления. Создание, отладка и тестирование инструментального программного обеспечения.

# 07.03.2023

**Тема:** Разработка концепции современной интегрированной среды разработки приложений. Технология компонентной разработки COM. Технология Java.

**Цель:** Изучить тему:Разработка концепции современной интегрированной среды разработки приложений. Технология компонентной разработки COM. Технология Java.

## Ход работы

### Дан двумерный массив действительных чисел (7 строк, 6 столбцов). Заполните его с клавиатуры. Поменяйте местами 3-й элемент 2-го столбца и 3-й элемент 5-й строки; подсчитайте количество чётных элементов; подсчитайте среднее арифметическое элементов 6-го столбца; выведите массив на экран до и после изменения.

#### Код:

using System;

class Program

{

static void Main(string[] args)

{

// Создаем пустой двумерный массив

double[,] array = new double[7, 6];

// Заполняем массив с клавиатуры

for (int i = 0; i < 7; i++)

{

for (int j = 0; j < 6; j++)

{

Console.Write($"Введите элемент [{i}, {j}]: ");

array[i, j] = double.Parse(Console.ReadLine());

}

}

// Выводим массив на экран до изменения

Console.WriteLine("Массив до изменения:");

for (int i = 0; i < 7; i++)

{

for (int j = 0; j < 6; j++)

{

Console.Write(array[i, j] + " ");

}

Console.WriteLine();

}

// Меняем местами элементы

double temp = array[2, 1];

array[2, 1] = array[4, 2];

array[4, 2] = temp;

// Подсчитываем количество чётных элементов

int evenCount = 0;

for (int i = 0; i < 7; i++)

{

for (int j = 0; j < 6; j++)

{

if (array[i, j] % 2 == 0)

{

evenCount++;

}

}

}

Console.WriteLine($"Количество четных элементов: {evenCount}");

// Подсчитываем среднее арифметическое элементов 6-го столбца

double sum = 0;

for (int i = 0; i < 7; i++)

{

sum += array[i, 5];

}

double average = sum / 7;

Console.WriteLine($"Среднее арифметическое элементов 6-го столбца: {average}");

// Выводим массив на экран после изменения

Console.WriteLine("Массив после изменения:");

for (int i = 0; i < 7; i++)

{

for (int j = 0; j < 6; j++)

{

Console.Write(array[i, j] + " ");

}

Console.WriteLine();

}

}

}

#### Консоль:

|  |
| --- |
| Рис. 15. Вид консоли |

**Вывод:** Изучил тему:Разработка концепции современной интегрированной среды разработки приложений. Технология компонентной разработки COM. Технология Java.

# 08.03.2023

**Тема:** Платформа .NET Framework. Общие сведения о CASE-средствах. Принципы построения и приемы работы с CASE-средствами.

**Цель:** Изучить тему:Платформа .NET Framework. Общие сведения о CASE-средствах. Принципы построения и приемы работы с CASE-средствами.

## Ход работы

### Элементы массива Х циклически сдвинуть на к позиций вправо. Результат напечатать в файл Excel.

#### Код:

using Excel = Microsoft.Office.Interop.Excel;

class Program

{

static void Main(string[] args)

{

// Инициализируем приложение Excel

Excel.Application app = new Excel.Application();

app.Visible = true;

// Создаем новую рабочую книгу и получаем ссылку на активный лист

Excel.Workbook workbook = app.Workbooks.Add();

Excel.Worksheet worksheet = (Excel.Worksheet)workbook.ActiveSheet;

// Задаем исходный массив

int[] X = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };

// Количество позиций для циклического сдвига

int k = 3;

// Выполняем циклический сдвиг элементов массива

int[] Y = new int[X.Length];

for (int i = 0; i < X.Length; i++)

{

int newIndex = (i + k) % X.Length;

Y[newIndex] = X[i];

}

// Выводим результаты в Excel

// Названия столбцов

worksheet.Cells[1, 1] = "Исходный массив X";

worksheet.Cells[1, 2] = "Циклически сдвинутый массив Y";

// Заполняем ячейки данными

for (int i = 0; i < X.Length; i++)

{

worksheet.Cells[i + 2, 1] = X[i];

worksheet.Cells[i + 2, 2] = Y[i];

}

// Сохраняем результат в файл

workbook.SaveAs(@"D:\техникум\уп\Результат.xlsx");

// Закрываем приложение Excel

app.Quit();

}

}

#### Результат работы программы:

|  |
| --- |
| Рис. 16. Результат работы программы |

**Вывод:** Изучил тему:Платформа .NET Framework. Общие сведения о CASE-средствах. Принципы построения и приемы работы с CASE-средствами.

# 09.03.2023

**Тема:** Основные функциональные возможности CASE-средствами. Классификация CASE-средств. Обзор современных CASE-средств.

**Цель:** Изучить тему:Основные функциональные возможности CASE-средствами. Классификация CASE-средств. Обзор современных CASE-средств.

## Ход работы

1. Создайте программу, показывающую скачущий мячик.

#### Код:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace \_09\_03\_2023\_1

{

public partial class Form1 : Form

{

private Panel ballPanel;

private Timer timer;

private int x = 100;

private int y = 100;

private int dx = 5;

private int dy = 5;

private int ballSize = 30;

public Form1()

{

InitializeComponent();

// Создаем панель, на которой будет отображаться мячик

ballPanel = new Panel();

ballPanel.BackColor = Color.White;

ballPanel.Size = new Size(400, 400);

ballPanel.Paint += new PaintEventHandler(DrawBall);

// Создаем таймер, который будет обновлять позицию мячика

timer = new Timer();

timer.Interval = 50;

timer.Tick += new EventHandler(UpdateBallPosition);

// Добавляем панель в форму

this.Controls.Add(ballPanel);

timer.Start();

}

private void DrawBall(object sender, PaintEventArgs e)

{

// Рисуем мячик на панели

e.Graphics.FillEllipse(Brushes.Red, x, y, ballSize, ballSize);

}

private void UpdateBallPosition(object sender, EventArgs e)

{

// Обновляем позицию мячика

x += dx;

y += dy;

// Если мячик касается верхней или нижней границы панели, меняем направление по y

if (y < 0 || y > ballPanel.Height - ballSize)

{

dy = -dy;

}

// Если мячик касается левой или правой границы панели, меняем направление по x

if (x < 0 || x > ballPanel.Width - ballSize)

{

dx = -dx;

}

// Обновляем отображение мячика на панели

ballPanel.Invalidate();

}

}

}

#### Форма:

|  |
| --- |
| Рис. 17. Вид формы |

**Вывод:** Изучил тему: Основные функциональные возможности CASE-средствами. Классификация CASE-средств. Обзор современных CASE-средств.

# 10.03.2023

**Тема:** Спецификация требований программного обеспечения. Управление документированием программного обеспечения. Оформление пакета документов по прохождение раздела учебной практики.

**Цель:** Изучить тему: Спецификация требований программного обеспечения. Управление документированием программного обеспечения. Оформление пакета документов по прохождение раздела учебной практики.

## Ход работы

### Написать метод, вычисляющий значение x2 + y2. С его помощью определить, с какой парой чисел (a, b) или (c, d) значение будет максимальным.

#### Код:

using System;

class Program

{

static void Main(string[] args)

{

int a = 1;

int b = 2;

int c = 3;

int d = 4;

int sum\_ab = CalcSumOfSquares(a, b);

int sum\_cd = CalcSumOfSquares(c, d);

if (sum\_ab > sum\_cd)

{

Console.WriteLine("Значение максимально для пары (a, b)");

}

else

{

Console.WriteLine("Значение максимально для пары (c, d)");

}

}

static int CalcSumOfSquares(int x, int y)

{

return x \* x + y \* y;

}

}

#### Вид консоли:

|  |
| --- |
| Рис. 18. Вид консоли |

**Вывод:** Изучил тему: Спецификация требований программного обеспечения. Управление документированием программного обеспечения. Оформление пакета документов по прохождение раздела учебной практики.

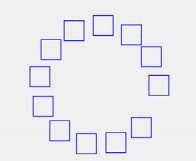
# 11.03.2023

**Тема:** Сборка и отладка программы в полном объеме, подготовка презентаций для защиты. Разработка пояснительной записки в соответствии с техническим заданием. Дифференцированный зачет.

**Цель:** Сборка и отладка программы в полном объеме, подготовка презентаций для защиты. Разработка пояснительной записки в соответствии с техническим заданием.

## Ход работы

### Напишите приложение, которое строит ряд квадратов, центры которых лежат на окружности. Число квадратов задаётся при первом вызове рекурсивного метода.



#### Код:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace \_11\_03\_2023

{

public partial class Form1 : Form

{

private int depth = 0;

private int maxDepth = 10;

private int squareSize = 40;

private int centerX;

private int centerY;

public Form1()

{

InitializeComponent();

centerX = ClientSize.Width / 2;

centerY = ClientSize.Height / 2;

}

private void Form1\_Paint(object sender, PaintEventArgs e)

{

DrawCircleSquares(e.Graphics, depth, centerX, centerY, squareSize);

}

private void DrawCircleSquares(Graphics g, int currentDepth, int x, int y, int size)

{

if (currentDepth > maxDepth) return;

// Draw the square

int halfSize = size / 2;

int left = x - halfSize;

int top = y - halfSize;

g.DrawRectangle(Pens.Black, left, top, size, size);

// Calculate the position of the next square

double angle = 2 \* Math.PI / maxDepth \* currentDepth;

int nextX = (int)(x + Math.Cos(angle) \* (halfSize + size));

int nextY = (int)(y + Math.Sin(angle) \* (halfSize + size));

// Recursively draw the next square

DrawCircleSquares(g, currentDepth + 1, nextX, nextY, size);

}

}

}

#### Вид формы:

|  |
| --- |
| Рис. 19. Вид формы |

**Вывод:** Изучена сборка и отладка программы в полном объеме, подготовка презентаций для защиты. Разработка пояснительной записки в соответствии с техническим заданием.

# Индивидуальное задание

**Тема:** Используя интегрированную среду разработки visual studio создать модель и протестировать работу программы.

**Цель:** Овладеть навыками использования интегрированной среды разработки visual studio, создать модель и протестировать работу программы.

**Задание:** Используя интегрированную среду разработки Visual Studio создать модель и протестировать работу программы.

1. Микрофронтенды: микросервисы для фронтенда
2. Преимущества и проблемы микрофронтендов
3. Проблемы проектирования микрофронтендов
4. Принципы построения микрофронтендов

## Ход работы

1. Изучение требований и анализ задачи: на этом этапе были изучены требования к сервису и определены основные задачи, которые должны быть решены при разработке системы.
2. Проектирование информационной системы: были проведены инфологическая и деталогическая моделирование, на основе которых была разработана структура базы данных и определены связи между сущностями. Для создания диаграмм используется draw.io.
3. Изучены архитектура, инструменты и особенности разработки приложений.
4. Создание макета: был создан макет главной страницы приложения с отражением на нем всех главных элементов.

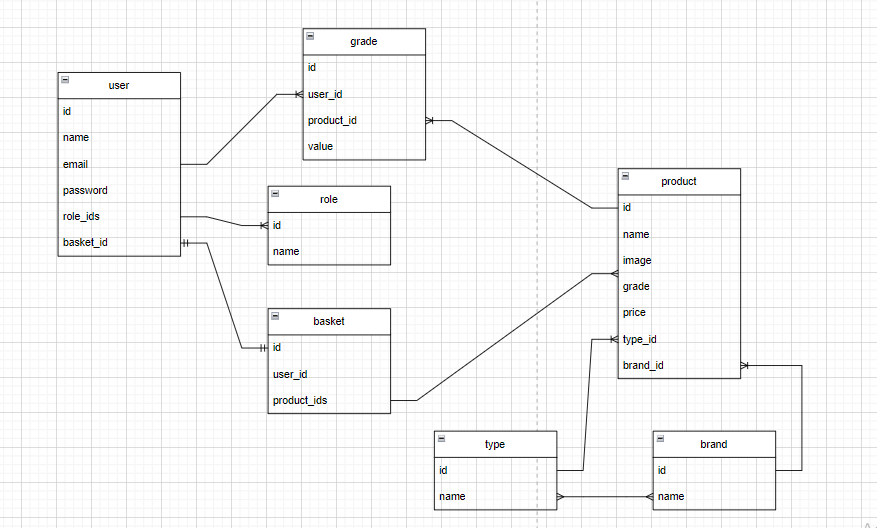


Рис. 20. Диаграмма БД

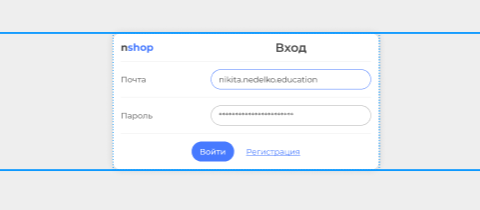


Рис. 21. Форма входа

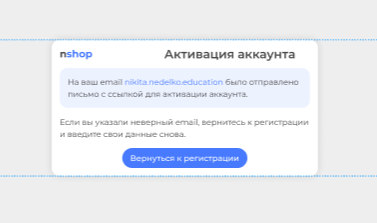


Рис. 22. Активация аккаунта

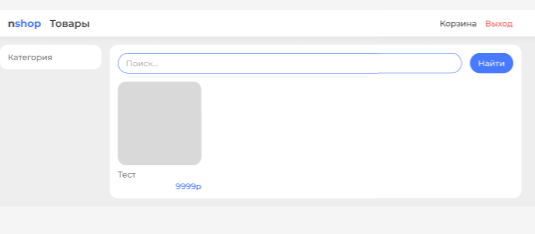


Рис. 23. Каталог товаров

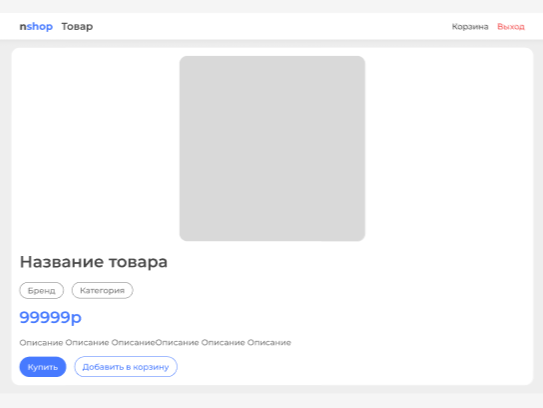


Рис. 24. Страница товара

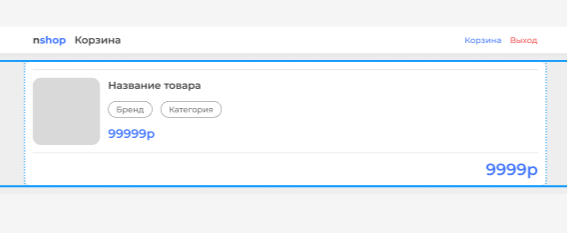


Рис. 25. Корзина

**Вывод:** В ходе выполнения индивидуального задания была создана диаграмма базы данных, также был разработан макет интернет магазина.