Министерство образования Республики Беларусь

Оршанский колледж ВГУ имени П.М.Машерова

**Отчет**

по учебной практике по программированию

«Модуль 2. Объектно-ориентированное программирование (ООП)»

по предмету «Конструирование программ и языки программирования»

Выполнил учащийся Казаченко Н.С.

группы 3ПОИС23 30.09.2025 г.

Проверил Алейников М.А.

30.09.2025 г.

Орша, 2025

**Цель:** закрепить знания о Объектно-ориентированном программировании (ООП).

**Модуль 2 Объектно-ориентированное программирование (ООП):**

**Общее задание для всех**

**Задача 1: Создание классов**

Создайте класс Person, представляющий человека. У этого класса должны быть поля для хранения имени, возраста и адреса. Добавьте методы для установки и получения значений этих полей. Затем создайте объекты этого класса и выведите информацию о них.

**Задача 2: Наследование и полиморфизм**

Создайте базовый класс Shape, представляющий геометрическую фигуру, и производные классы Circle и Rectangle. В базовом классе определите метод Area(), который возвращает площадь фигуры, и метод Perimeter(), который возвращает периметр фигуры. В производных классах переопределите эти методы для соответствующих фигур (круг и прямоугольник). Создайте объекты всех классов и выведите их площади и периметры.

**Задача 3: Композиция**

Создайте классы Author и Book. Класс Author должен содержать информацию об авторе (имя и год рождения). Класс Book должен содержать информацию о книге (название, год выпуска и автора). Используйте композицию, чтобы связать объекты Author и Book. Создайте несколько объектов Author и Book и выведите информацию о них.

**Задача 4: Интерфейсы и абстрактные классы**

Создайте интерфейс IDrawable с методом Draw(), который выводит информацию о рисуемом объекте. Создайте классы Circle, Rectangle и Triangle, реализующие этот интерфейс. Создайте массив объектов, реализующих интерфейс IDrawable, и вызовите метод Draw() для каждого из них.

**Задача 5: События**

Создайте класс TemperatureSensor, который генерирует событие TemperatureChanged, когда измеренная температура меняется. Создайте класс Thermostat, который подписывается на событие TemperatureChanged и реагирует на изменение температуры, включая или выключая отопление.

**Вариант 2.**

1. Создайте класс "Автомобиль" с полями: марка, модель, год выпуска и цена. Реализуйте методы для расчета стоимости автомобиля с учетом скидки и налога на добавленную стоимость.
2. Создайте структуру с именем student, содержащую поля: фамилия и инициалы, номер группы, успеваемость (массив из пяти элементов). Создать массив из десяти элементов такого типа, упорядочить записи по возрастанию среднего балла. Добавить возможность вывода фамилий и номеров групп студентов, имеющих оценки, равные только 4 или 5.
3. Создание класса "Геометрическая фигура" (Shape) и его производных: Создайте абстрактный класс "Геометрическая фигура" с методом для вычисления площади. Создайте производные классы, например, "Круг" (Circle), "Прямоугольник" (Rectangle), "Треугольник" (Triangle) и т. д. Реализуйте методы для вычисления площади у каждого класса. Создайте объекты различных классов и протестируйте их функциональность.

**Ход работы**

**Выполнение задания 1**

**Листинг 1.** Классы программы

using System;

namespace Задание\_1

{

class Person

{

public string Name { get; set; }

public int Age { get; set; }

public string Address { get; set; }

// Конструктор класса Person

public Person(string name, int age, string address)

{

Name = name;

Age = age;

Address = address;

}

// Метод для вывода информации о человеке

public void Print()

{

Console.WriteLine($"Имя: {Name}, Возраст: {Age}, Адрес: {Address}");

}

}

}

**Листинг 2.** Код программы

using System;

namespace Задание\_1

{

class Program

{

static void Main(string[] args)

{

// Ввод данных о первом человеке

Console.WriteLine("Введите данные о первом человеке: ");

Console.Write("Имя: ");

string Name1 = Console.ReadLine();

Console.Write("Возраст: ");

int Age1 = int.Parse(Console.ReadLine());

Console.Write("Адрес: ");

string Address1 = Console.ReadLine();

// Создание объекта класса Person с введёнными данными

Person person1 = new Person(Name1, Age1, Address1);

// Ввод данных о втором человеке

Console.WriteLine("Введите данные о втором человеке: ");

Console.Write("Имя: ");

string Name2 = Console.ReadLine();

Console.Write("Возраст: ");

int Age2 = int.Parse(Console.ReadLine());

Console.Write("Адрес: ");

string Address2 = Console.ReadLine();

// Создание второго объекта класса Person с введёнными данными

Person person2 = new Person(Name2, Age2, Address2);

// Вывод информации о первом человеке

Console.WriteLine("\nИнформация о первом человеке:");

person1.Print();

// Вывод информации о втором человеке

Console.WriteLine("\nИнформация о втором человеке:");

person2.Print();

}

}

}

На рисунке 1 представлено выполнение задания создания класса Person с полями для хранения имени, возраста и адреса

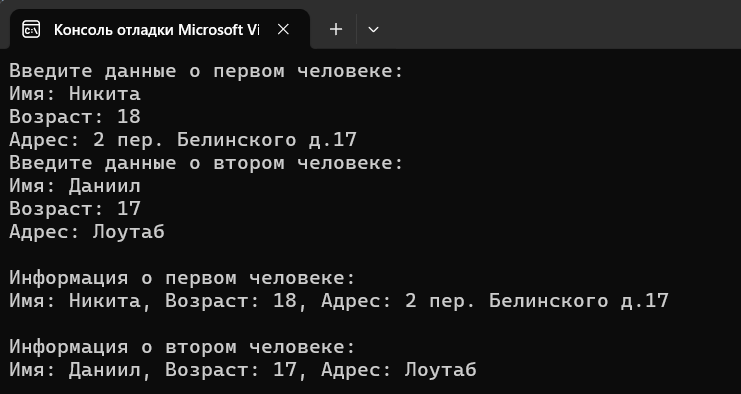


Рисунок 1 – Выполнение работы программы

**Выполнение задания 2**

**Листинг 3.** Классы программы

using System;

namespace Задание\_2

{

class Shape

{

public virtual double Area { get; set; }

public virtual double Perimeter { get; set; }

public virtual void Print()

{

Console.WriteLine($"Площадь: {Area:F2}, Периметр: {Perimeter:F2}");

}

}

class Circle : Shape

{

public double Radius { get; set; }

public Circle(double radius)

{

Radius = radius;

Area = Math.PI \* Radius \* Radius;

Perimeter = 2 \* Math.PI \* Radius;

}

public override void Print()

{

Console.WriteLine($"Круг с радиусом: {Radius}");

base.Print();

}

}

class Rectangle : Shape

{

public double Width { get; set; }

public double Height { get; set; }

public Rectangle(double width, double height)

{

Width = width;

Height = height;

Area = Width \* Height;

Perimeter = 2 \* (Width + Height);

}

public override void Print()

{

Console.WriteLine($"Прямоугольник с шириной: {Width} и высотой: {Height}");

base.Print();

}

}

}

**Листинг 4.** Код программы

using System;

namespace Задание\_2

{

class Program

{

static void Main(string[] args)

{

// Ввод данных о круге

Console.WriteLine("Введите данные о круге: ");

Console.Write("Радиус: ");

double radius = double.Parse(Console.ReadLine());

Circle circle = new Circle(radius);

// Ввод данных о прямоугольнике

Console.WriteLine("Введите данные о прямоугольнике: ");

Console.Write("Ширина: ");

double width = double.Parse(Console.ReadLine());

Console.Write("Высота: ");

double height = double.Parse(Console.ReadLine());

Rectangle rectangle = new Rectangle(width, height);

// Вывод данных о фигурах

Console.WriteLine("\nИнформация о круге:");

circle.Print();

Console.WriteLine("\nИнформация о прямоугольнике:");

rectangle.Print();

}

}

}

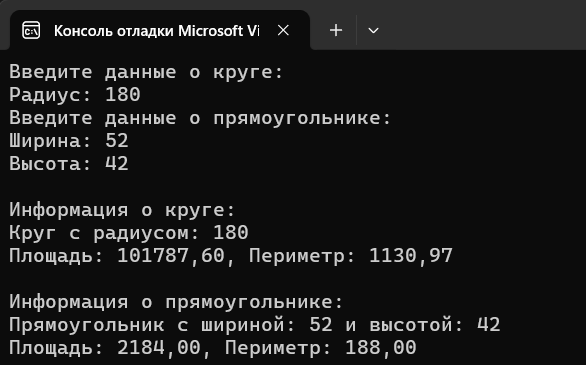
На рисунке 2 представлено выполнение задания создания базового класса Shape с методами для вычисления площади и периметра фигуры 

Рисунок 2 – Выполнение работы программы

**Выполнение задания 3**

**Листинг 5.** Классы программы

using System;

namespace Задание\_3

{

class Author

{

// Свойства для имени и года рождения автора

public string Name { get; set; }

public int BirthYear { get; set; }

// Конструктор для инициализации имени и года рождения автора

public Author(string name, int birthYear)

{

Name = name;

BirthYear = birthYear;

}

// Метод для вывода информации об авторе

public void Print()

{

Console.WriteLine($"Автор: {Name}, Год рождения: {BirthYear}");

}

}

class Book

{

// Свойства для названия книги, года выпуска и автора

public string Title { get; set; }

public int ReleaseYear { get; set; }

public Author Author { get; set; } // Композиция: объект класса Author включен в объект Book

// Конструктор для инициализации книги с названием, годом выпуска и объектом автора

public Book(string title, int releaseYear, Author author)

{

Title = title;

ReleaseYear = releaseYear;

Author = author;

}

// Метод для вывода информации о книге

public void Print()

{

Console.WriteLine($"Книга: {Title}, Год выпуска: {ReleaseYear}");

Author.Print(); // Вывод информации об авторе через объект Author

}

}

}

**Листинг 6.** Код программы

using System;

namespace Задание\_3

{

class Program

{

static void Main(string[] args)

{

// Ввод данных об авторе

Console.WriteLine("Введите информацию о первом авторе:");

Console.Write("Имя автора: ");

string authorName1 = Console.ReadLine();

Console.Write("Год рождения автора: ");

int authorBirthYear1 = int.Parse(Console.ReadLine());

Author author1 = new Author(authorName1, authorBirthYear1); // Создание объекта автора

Console.WriteLine("Введите информацию о книге:");

Console.Write("Название книги: ");

string bookTitle1 = Console.ReadLine();

Console.Write("Год выпуска книги: ");

int bookReleaseYear1 = int.Parse(Console.ReadLine());

// Создание объекта книги с введённым автором

Book book1 = new Book(bookTitle1, bookReleaseYear1, author1);

// Ввод данных для второго автора и книги

Console.WriteLine("\nВведите информацию о втором авторе:");

Console.Write("Имя автора: ");

string authorName2 = Console.ReadLine();

Console.Write("Год рождения автора: ");

int authorBirthYear2 = int.Parse(Console.ReadLine());

Author author2 = new Author(authorName2, authorBirthYear2); // Создание второго объекта автора

Console.WriteLine("Введите информацию о второй книге:");

Console.Write("Название книги: ");

string bookTitle2 = Console.ReadLine();

Console.Write("Год выпуска книги: ");

int bookReleaseYear2 = int.Parse(Console.ReadLine());

// Создание второго объекта книги

Book book2 = new Book(bookTitle2, bookReleaseYear2, author2);

// Вывод информации о книгах и авторах

Console.WriteLine("\nИнформация о первой книге:");

book1.Print();

Console.WriteLine("\nИнформация о второй книге:");

book2.Print();

}

}

}

На рисунке 3 представлено выполнение задания создания класса Author, содержащего информацию об авторе (имя и год рождения)

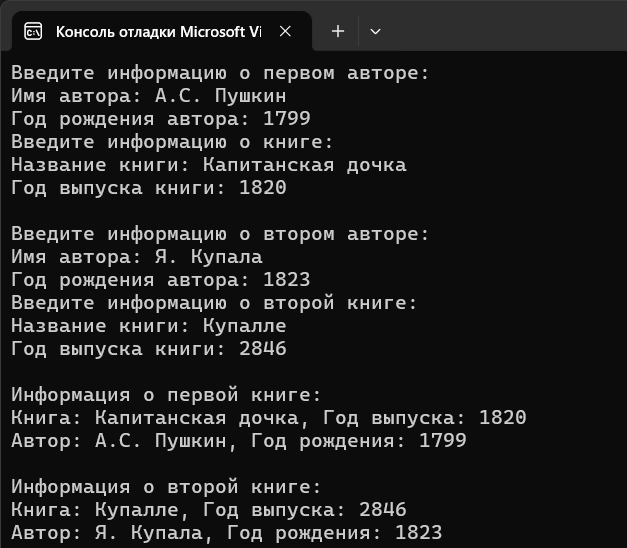


Рисунок 3 – Выполнение работы программы

**Выполнение задания 4**

**Листинг 7.** Классы программы

using System;

namespace Задание\_4

{

// Определение интерфейса IDrawable

interface IDrawable

{

void Draw();

}

// Класс Circle, реализующий интерфейс IDrawable

class Circle : IDrawable

{

public double Radius { get; set; }

public Circle(double radius)

{

Radius = radius;

}

// Реализация метода Draw для круга

public void Draw()

{

Console.WriteLine($"Рисуем круг с радиусом: {Radius}");

}

}

// Класс Rectangle, реализующий интерфейс IDrawable

class Rectangle : IDrawable

{

public double Width { get; set; }

public double Height { get; set; }

public Rectangle(double width, double height)

{

Width = width;

Height = height;

}

// Реализация метода Draw для прямоугольника

public void Draw()

{

Console.WriteLine($"Рисуем прямоугольник с шириной: {Width} и высотой: {Height}");

}

}

// Класс Triangle, реализующий интерфейс IDrawable

class Triangle : IDrawable

{

public double SideA { get; set; }

public double SideB { get; set; }

public double SideC { get; set; }

public Triangle(double sideA, double sideB, double sideC)

{

SideA = sideA;

SideB = sideB;

SideC = sideC;

}

// Реализация метода Draw для треугольника

public void Draw()

{

е Console.WriteLine($"Рисуем треугольник со сторонами: {SideA}, {SideB}, {SideC}");

}

}

}

**Листинг 8.** Код программы

using System;

namespace Задание\_4

{

class Program

{

static void Main(string[] args)

{

// Создание массива объектов, реализующих интерфейс IDrawable

IDrawable[] shapes = new IDrawable[3];

// Ввод данных для круга

Console.Write("Введите радиус круга:");

double radius = double.Parse(Console.ReadLine());

shapes[0] = new Circle(radius);

// Ввод данных для прямоугольника

Console.Write("Введите ширину прямоугольника:");

double width = double.Parse(Console.ReadLine());

Console.Write("Введите высоту прямоугольника:");

double height = double.Parse(Console.ReadLine());

shapes[1] = new Rectangle(width, height);

// Ввод данных для треугольника

Console.Write("Введите длину первой стороны треугольника:");

double sideA = double.Parse(Console.ReadLine());

Console.Write("Введите длину второй стороны треугольника:");

double sideB = double.Parse(Console.ReadLine());

Console.Write("Введите длину третьей стороны треугольника:");

double sideC = double.Parse(Console.ReadLine());

shapes[2] = new Triangle(sideA, sideB, sideC);

// Вызов метода Draw для каждого объекта

Console.WriteLine("\nРисуем фигуры:");

foreach (IDrawable shape in shapes)

{

shape.Draw();

}

}

}

}

На рисунке 4 представлено выполнение задания создания интерфейса IDrawable с методом Draw(), который выводит информацию о рисуемом объекте

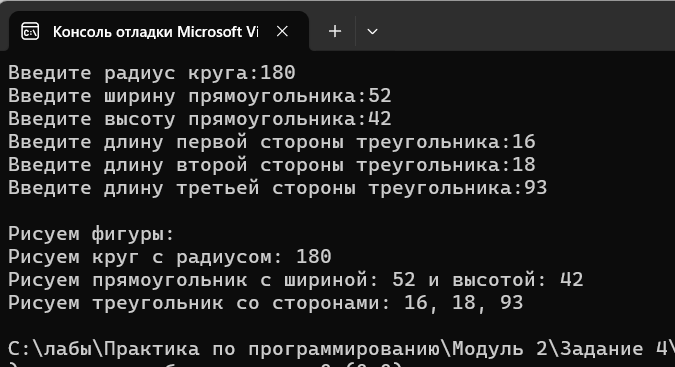


Рисунок 4 – Выполнение работы программы

**Выполнение задания 5**

**Листинг 9.** Классы программы

using System;

namespace Задание\_5

{

class TemperatureSensor

{

// Делегат для события изменения температуры

public delegate void TemperatureChangedHandler(double newTemperature);

// Событие на основе делегата

public event TemperatureChangedHandler TemperatureChanged;

private double \_temperature;

// Свойство для установки и получения температуры

public double Temperature

{

get { return \_temperature; }

set

{

if (\_temperature != value)

{

\_temperature = value;

OnTemperatureChanged(\_temperature); // Генерация события при изменении

}

}

}

// Метод, вызываемый при изменении температуры, для генерации события

protected virtual void OnTemperatureChanged(double newTemperature)

{

if (TemperatureChanged != null)

{

TemperatureChanged(newTemperature); // Если есть подписчики, вызываем событие

}

}

}

class Thermostat

{

private double \_desiredTemperature;

// Конструктор для задания желаемой температуры

public Thermostat(double desiredTemperature)

{

\_desiredTemperature = desiredTemperature;

}

// Метод, обрабатывающий событие изменения температуры

public void OnTemperatureChanged(double newTemperature)

{

if (newTemperature < \_desiredTemperature)

{

Console.WriteLine($"Температура {newTemperature}°C слишком низкая. Включение отопления...");

}

else if (newTemperature > \_desiredTemperature)

{

Console.WriteLine($"Температура {newTemperature}°C слишком высокая. Выключение отопления...");

}

else

{

Console.WriteLine($"Температура {newTemperature}°C оптимальная. Оставляем текущее состояние отопления.");

}

}

}

}

**Листинг 10.** Код программы

using System;

namespace Задание\_5

{

class Program

{

static void Main(string[] args)

{

// Создание объекта датчика температуры

TemperatureSensor sensor = new TemperatureSensor();

// Ввод желаемой температуры для термостата

Console.WriteLine("Введите желаемую температуру для термостата:");

double desiredTemperature = double.Parse(Console.ReadLine());

// Создание объекта термостата

Thermostat thermostat = new Thermostat(desiredTemperature);

// Подписка термостата на событие изменения температуры

sensor.TemperatureChanged += thermostat.OnTemperatureChanged;

// Ввод данных для изменения температуры

Console.WriteLine("Введите текущую температуру:");

double currentTemperature = double.Parse(Console.ReadLine());

sensor.Temperature = currentTemperature;

// Ввод следующего значения температуры

Console.WriteLine("Введите новую температуру:");

double newTemperature = double.Parse(Console.ReadLine());

sensor.Temperature = newTemperature;

}

}

}

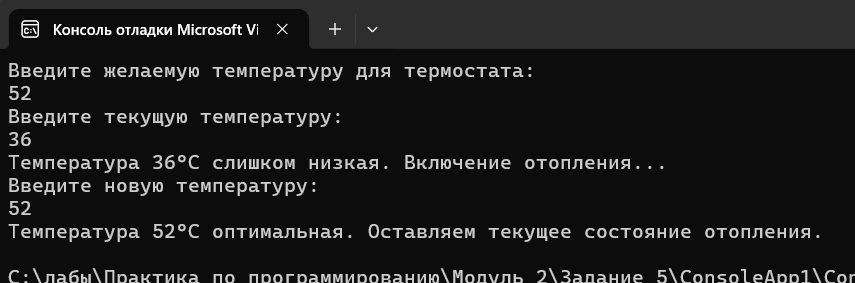
На рисунке 1 представлено выполнение задания создания класса TemperatureSensor, генерирующего событие TemperatureChanged при изменении температуры

Рисунок 5 – Выполнение работы программы

**Выполнение задания 1 (Вариант 2)**

**Листинг 11.** Классы программы

using System;

namespace Задание\_1

{

class Car

{

public string brand { get; set; }

public string model { get; set; }

public int year\_of\_release { get; set; }

public int price { get; set; }

public Car(string brand, string model, int year\_of\_release, int price)

{

this.brand = brand;

this.model = model;

this.year\_of\_release = year\_of\_release;

this.price = price;

}

// Метод для вывода информации о машине

public void Print()

{

Console.WriteLine($"Марка: {brand}, Модель: {model}, Год выпуска: {year\_of\_release}, Цена: {price}");

}

// Метод для расчета стоимости с учетом скидки

public double GetPriceWithDiscount(double discountPercentage)

{

double discountAmount = price \* (discountPercentage / 100);

return price - discountAmount;

}

// Метод для расчета стоимости с учетом НДС

public double GetPriceWithTax(double taxPercentage)

{

double taxAmount = price \* (taxPercentage / 100);

return price + taxAmount;

}

}

}

**Листинг 12.** Код программы

using System;

namespace Задание\_1

{

class Program

{

static void Main(String[] args)

{

Console.WriteLine("Введите информацию о первой машине: ");

Console.Write("Марка: ");

string brand1 = Console.ReadLine();

Console.Write("Модель: ");

string model1 = Console.ReadLine();

Console.Write("Год выпуска: ");

int year\_of\_release1 = int.Parse(Console.ReadLine());

Console.Write("Цена: ");

int price1 = int.Parse(Console.ReadLine());

Car car1 = new Car(brand1, model1, year\_of\_release1, price1);

Console.WriteLine("Введите информацию о второй машине: ");

Console.Write("Марка: ");

string brand2 = Console.ReadLine();

Console.Write("Модель: ");

string model2 = Console.ReadLine();

Console.Write("Год выпуска: ");

int year\_of\_release2 = int.Parse(Console.ReadLine());

Console.Write("Цена: ");

int price2 = int.Parse(Console.ReadLine());

Car car2 = new Car(brand2, model2, year\_of\_release2, price2);

// Выводим информацию о машинах

Console.WriteLine("\nИнформация о первой машине:");

car1.Print();

Console.WriteLine("\nИнформация о второй машине:");

car2.Print();

// Рассчет стоимости с учетом скидки и НДС

Console.WriteLine("\nВведите процент скидки: ");

double discount = double.Parse(Console.ReadLine());

Console.WriteLine("Введите процент налога на добавленную стоимость (НДС): ");

double tax = double.Parse(Console.ReadLine());

// Рассчет для первой машины

double priceWithDiscount1 = car1.GetPriceWithDiscount(discount);

double priceWithTax1 = car1.GetPriceWithTax(tax);

Console.WriteLine($"\nЦена первой машины с учетом скидки: {priceWithDiscount1}");

Console.WriteLine($"Цена первой машины с учетом НДС: {priceWithTax1}");

// Рассчет для второй машины

double priceWithDiscount2 = car2.GetPriceWithDiscount(discount);

double priceWithTax2 = car2.GetPriceWithTax(tax);

Console.WriteLine($"\nЦена второй машины с учетом скидки: {priceWithDiscount2}");

Console.WriteLine($"Цена второй машины с учетом НДС: {priceWithTax2}");

}

}

}

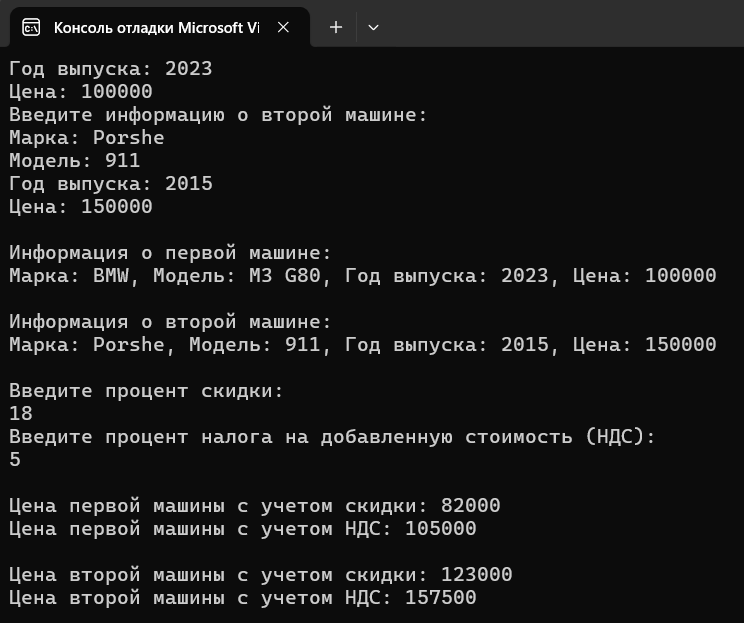
На рисунке 6 представлено выполнение задания создания класса Автомобиль с полями: марка, модель, год выпуска и цена.

Рисунок 6 – Выполнение работы программы

**Выполнение задания 2 (Вариант 2)**

**Листинг 13.** Классы программы

using System;

namespace Задание\_2

{

public struct Student

{

public string LastName { get; set; } // Фамилия студента

public string Initials { get; set; } // Инициалы студента

public int GroupNumber { get; set; } // Номер группы

public int[] Grades { get; set; } // Успеваемость (оценки)

// Метод для расчета среднего балла

public double GetAverageGrade()

{

return Grades.Average(); // Возвращаем среднее значение из массива оценок

}

// Метод для проверки, что все оценки равны 4 или 5

public bool HasOnlyHighGrades()

{

return Grades.All(grade => grade == 4 || grade == 5);

}

}

}

**Листинг 14.** Код программы

using System;

namespace Задание\_2

{

// Program.cs

using System;

class Program

{

static void Main(string[] args)

{

// Создаем массив из 10 студентов

Student[] students = new Student[10];

// Ввод данных для каждого студента

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

{

Console.WriteLine($"Введите данные для студента №{i + 1}:");

Console.Write("Фамилия: ");

string lastName = Console.ReadLine();

Console.Write("Инициалы: ");

string initials = Console.ReadLine();

Console.Write("Номер группы: ");

int groupNumber = int.Parse(Console.ReadLine());

int[] grades = new int[5];

Console.WriteLine("Введите 5 оценок:");

for (int j = 0; j < grades.Length; j++)

{

Console.Write($"Оценка {j + 1}: ");

grades[j] = int.Parse(Console.ReadLine());

}

// Инициализация студента

students[i] = new Student

{

LastName = lastName,

Initials = initials,

GroupNumber = groupNumber,

Grades = grades

};

}

// Сортировка студентов по возрастанию среднего балла

var sortedStudents = students.OrderBy(s => s.GetAverageGrade()).ToArray();

// Вывод отсортированных студентов

Console.WriteLine("\nСтуденты, отсортированные по возрастанию среднего балла:");

foreach (var student in sortedStudents)

{

Console.WriteLine($"{student.LastName} ({student.Initials}), группа: {student.GroupNumber}, средний балл: {student.GetAverageGrade():F2}");

}

// Вывод студентов, у которых все оценки равны 4 или 5

Console.WriteLine("\nСтуденты с оценками только 4 или 5:");

foreach (var student in sortedStudents)

{

if (student.HasOnlyHighGrades())

{

Console.WriteLine($"{student.LastName} ({student.Initials}), группа: {student.GroupNumber}");

}

}

}

}

}

На рисунке 7 представлено выполнение задания создания структуры student с полями: фамилия и инициалы, номер группы и массив успеваемости из пяти элементов.

Рисунок 7 – Выполнение работы программы

**Выполнение задания 3 (Вариант 2)**

**Листинг 15.** Классы программы

using System;

// Абстрактный класс "Геометрическая фигура"

public abstract class Shape

{

// Абстрактный метод для вычисления площади

public abstract double CalculateArea();

}

// Класс "Круг"

public class Circle : Shape

{

public double Radius { get; set; }

// Конструктор для инициализации радиуса круга

public Circle(double radius)

{

Radius = radius;

}

// Реализация метода для вычисления площади круга

public override double CalculateArea()

{

return Math.PI \* Radius \* Radius;

}

}

// Класс "Прямоугольник"

public class Rectangle : Shape

{

public double Width { get; set; }

public double Height { get; set; }

// Конструктор для инициализации ширины и высоты прямоугольника

public Rectangle(double width, double height)

{

Width = width;

Height = height;

}

// Реализация метода для вычисления площади прямоугольника

public override double CalculateArea()

{

return Width \* Height;

}

}

// Класс "Треугольник"

public class Triangle : Shape

{

public double BaseLength { get; set; }

public double Height { get; set; }

// Конструктор для инициализации основания и высоты треугольника

public Triangle(double baseLength, double height)

{

BaseLength = baseLength;

Height = height;

}

// Реализация метода для вычисления площади треугольника

public override double CalculateArea()

{

return 0.5 \* BaseLength \* Height;

}

}

**Листинг 16.** Код программы

using System;

namespace Задание\_3

{

class Program

{

static void Main(string[] args)

{

// Создание объекта круга

Console.Write("Введите радиус круга: ");

double radius = double.Parse(Console.ReadLine());

Circle circle = new Circle(radius);

// Создание объекта прямоугольника

Console.Write("Введите ширину прямоугольника: ");

double width = double.Parse(Console.ReadLine());

Console.Write("Введите высоту прямоугольника: ");

double height = double.Parse(Console.ReadLine());

Rectangle rectangle = new Rectangle(width, height);

// Создание объекта треугольника

Console.Write("Введите основание треугольника: ");

double baseLength = double.Parse(Console.ReadLine());

Console.Write("Введите высоту треугольника: ");

double triangleHeight = double.Parse(Console.ReadLine());

Triangle triangle = new Triangle(baseLength, triangleHeight);

// Вывод информации о площадях фигур

Console.WriteLine($"\nПлощадь круга: {circle.CalculateArea():F2}");

Console.WriteLine($"Площадь прямоугольника: {rectangle.CalculateArea():F2}");

Console.WriteLine($"Площадь треугольника: {triangle.CalculateArea():F2}");

}

}

}

Работа программы представлена на рисунке 8

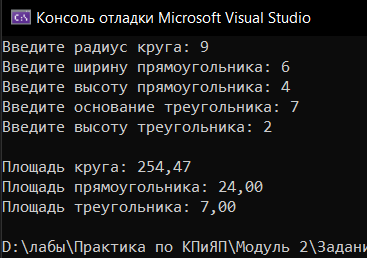


Рисунок 8 – Работа программы на ЯП С#

**Ссылка на репозиторий:** [Asevzz/Module2: Практика КПиЯП (github.com)](https://github.com/Asevzz/Module2)

**Вывод:** в ходе выполнения практического задания (модуль 2) были закреплены базовые понятия ООП.