Министерство образования Республики Беларусь

Оршанский колледж ВГУ имени П.М. Машерова

Отчет

По модулю №4

«Задания по интерфейсам»

по учебной практике по программированию

Выполнил учащийся Войтеховская К.В.

группа 3ПОИС23 29.09.2025 г.

Проверил Алейников М.А.

30.09.2025 г.

Орша, 2025

**Ход работы.**

**Выполнение практических заданий.**

**Выполнение задания 1.** Создайте интерфейс "Фигура" с методами для вычисления площади и периметра. Затем реализуйте этот интерфейс в классах геометрических фигур (круг, прямоугольник, треугольник).

Листинг 1. Код программы с интерфейсом «Фигура»

using System;

// Интерфейс

interface IShape

{

double GetArea();

double GetPerimeter();

}

// Круг

class Circle : IShape

{

public double Radius;

public double GetArea()

{

return 3.14 \* Radius \* Radius;

}

public double GetPerimeter()

{

return 2 \* 3.14 \* Radius;

}

}

// Прямоугольник

class Rectangle : IShape

{

public double Width;

public double Height;

public double GetArea()

{

return Width \* Height;

}

public double GetPerimeter()

{

return 2 \* (Width + Height);

}

}

// Треугольник

class Triangle : IShape

{

public double Side1;

public double Side2;

public double Side3;

public double GetArea()

{

return 0.5 \* Side1 \* Side2;

}

public double GetPerimeter()

{

return Side1 + Side2 + Side3;

}

}

class Program

{

static void Main()

{

Console.WriteLine("Фигуры:");

// Создание круга

Circle circle = new Circle();

circle.Radius = 5;

Console.WriteLine($"Круг: площадь = {circle.GetArea()}, периметр = {circle.GetPerimeter()}");

// Создание прямоугольника

Rectangle rect = new Rectangle();

rect.Width = 4;

rect.Height = 6;

Console.WriteLine($"Прямоугольник: площадь = {rect.GetArea()}, периметр = {rect.GetPerimeter()}");

// Создание треугольника

Triangle triangle = new Triangle();

triangle.Side1 = 3;

triangle.Side2 = 4;

triangle.Side3 = 5;

Console.WriteLine($"Треугольник: площадь = {triangle.GetArea()}, периметр = {triangle.GetPerimeter()}");

}

}

На рисунке 1 представлен результат выполнения программы.

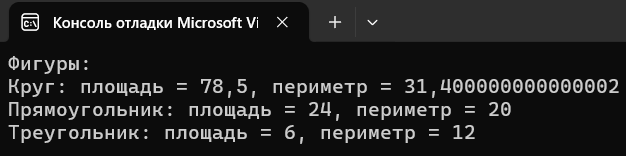


Рисунок 1 – Вывод параметров фигур

**Выполнение задания 2.** Разработайте приложение для учёта продуктов в магазине с использованием интерфейсов. Создайте интерфейс "Товар" с методами для определения стоимости и остатка товара на складе. Реализуйте этот интерфейс в классах различных товаров.

Листинг 2. Код программы приложения учёта товара

using System;

// Интерфейс товара

interface IProduct

{

double GetPrice();

int GetStock();

}

// Класс продукта

class Product : IProduct

{

public string Name;

public double Price;

public int Stock;

public double GetPrice()

{

return Price;

}

public int GetStock()

{

return Stock;

}

}

// Класс продукта со скидкой

class DiscountProduct : IProduct

{

public string Name;

public double Price;

public int Stock;

public double Discount; // скидка в процентах

public double GetPrice()

{

return Price \* (1 - Discount / 100);

}

public int GetStock()

{

return Stock;

}

}

class Program

{

static void Main()

{

Console.WriteLine("Товары в магазине:");

// Обычный товар

Product milk = new Product();

milk.Name = "Молоко";

milk.Price = 80;

milk.Stock = 10;

Console.WriteLine($"{milk.Name}: цена = {milk.GetPrice()}, остаток = {milk.GetStock()}");

// Товар со скидкой

DiscountProduct bread = new DiscountProduct();

bread.Name = "Хлеб";

bread.Price = 50;

bread.Stock = 5;

bread.Discount = 20; // 20% скидка

Console.WriteLine($"{bread.Name}: цена со скидкой = {bread.GetPrice()}, остаток = {bread.GetStock()}");

// Ещё один товар

Product cheese = new Product();

cheese.Name = "Сыр";

cheese.Price = 300;

cheese.Stock = 3;

Console.WriteLine($"{cheese.Name}: цена = {cheese.GetPrice()}, остаток = {cheese.GetStock()}");

}

}

На рисунке 2 представлен результат выполнения программы.

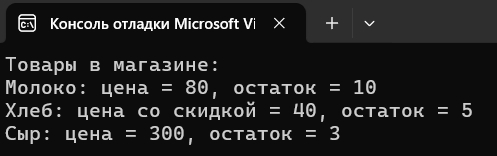


Рисунок 2 – Вывод информации о товарах

**Выполнение задания 3.** Создайте систему учёта студентов в университете с помощью интерфейсов. Создайте интерфейс "Студент" с методами для определения среднего балла и получения информации о курсе. Реализуйте этот интерфейс в классах студентов разных курсов.

Листинг 3. Код программы учёта студентов

using System;

// Интерфейс студента

interface IStudent

{

double GetAverageGrade();

string GetCourseInfo();

}

// Студент бакалавриата

class BachelorStudent : IStudent

{

public string Name;

public int Course;

public double[] Grades;

public double GetAverageGrade()

{

double sum = 0;

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

{

sum += Grades[i];

}

return sum / Grades.Length;

}

public string GetCourseInfo()

{

return $"Бакалавр {Course} курса";

}

}

// Студент магистратуры

class MasterStudent : IStudent

{

public string Name;

public int Course;

public double[] Grades;

public double GetAverageGrade()

{

double sum = 0;

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

{

sum += Grades[i];

}

return sum / Grades.Length;

}

public string GetCourseInfo()

{

return $"Магистр {Course} курса";

}

}

class Program

{

static void Main()

{

Console.WriteLine("Студенты:");

// Студент бакалавриата

BachelorStudent student1 = new BachelorStudent();

student1.Name = "Александр";

student1.Course = 2;

student1.Grades = new double[] { 3, 7, 8, 2 };

Console.WriteLine($"{student1.Name}: {student1.GetCourseInfo()}, средний балл = {student1.GetAverageGrade():F2}");

// Студент магистратуры

MasterStudent student2 = new MasterStudent();

student2.Name = "Светлана";

student2.Course = 1;

student2.Grades = new double[] { 10, 8, 9, 10 };

Console.WriteLine($"{student2.Name}: {student2.GetCourseInfo()}, средний балл = {student2.GetAverageGrade():F2}");

// Ещё один студент

BachelorStudent student3 = new BachelorStudent();

student3.Name = "Алексей";

student3.Course = 4;

student3.Grades = new double[] { 8, 7, 6, 7.1 };

Console.WriteLine($"{student3.Name}: {student3.GetCourseInfo()}, средний балл = {student3.GetAverageGrade():F2}");

}

}

На рисунке 3 представлен результат выполнения программы.

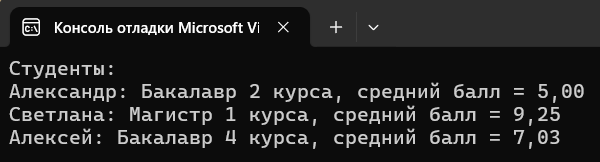


Рисунок 3 – Вывод информации о студентах

**Выполнение задания 4.** Реализуйте приложение для работы с библиотекой книг с использованием интерфейсов. Создайте интерфейс "Книга" с методами для проверки доступности и выдачи книги. Реализуйте этот интерфейс в классах различных книг.

Листинг 4. Код программы приложения работой с библиотекой книг

using System;

// Интерфейс книги

interface IBook

{

bool IsAvailable();

void Borrow();

void Return();

}

// Обычная книга

class Book : IBook

{

public string Title;

public string Author;

public bool IsBorrowed;

public bool IsAvailable()

{

return !IsBorrowed;

}

public void Borrow()

{

if (!IsBorrowed)

{

IsBorrowed = true;

Console.WriteLine($"Книга '{Title}' выдана");

}

else

{

Console.WriteLine($"Книга '{Title}' уже выдана");

}

}

public void Return()

{

if (IsBorrowed)

{

IsBorrowed = false;

Console.WriteLine($"Книга '{Title}' возвращена");

}

else

{

Console.WriteLine($"Книга '{Title}' уже в библиотеке");

}

}

}

// Учебник

class Textbook : IBook

{

public string Title;

public string Author;

public string Subject;

public bool IsBorrowed;

public bool IsAvailable()

{

return !IsBorrowed;

}

public void Borrow()

{

if (!IsBorrowed)

{

IsBorrowed = true;

Console.WriteLine($"Учебник '{Title}' по предмету {Subject} выдан");

}

else

{

Console.WriteLine($"Учебник '{Title}' уже выдан");

}

}

public void Return()

{

if (IsBorrowed)

{

IsBorrowed = false;

Console.WriteLine($"Учебник '{Title}' возвращен");

}

else

{

Console.WriteLine($"Учебник '{Title}' уже в библиотеке");

}

}

}

class Program

{

static void Main()

{

Console.WriteLine("Библиотека:");

// Обычная книга

Book book1 = new Book();

book1.Title = "Кровавая корона";

book1.Author = "Шуи Цянчэн";

book1.IsBorrowed = false;

// Учебник

Textbook textbook1 = new Textbook();

textbook1.Title = "Математика";

textbook1.Author = "Иванов";

textbook1.Subject = "Математика";

textbook1.IsBorrowed = false;

// Проверяем доступность

Console.WriteLine($"Книга '{book1.Title}' доступна: {book1.IsAvailable()}");

Console.WriteLine($"Учебник '{textbook1.Title}' доступен: {textbook1.IsAvailable()}");

// Выдаем книги

book1.Borrow();

textbook1.Borrow();

// Пытаемся выдать снова

book1.Borrow();

// Возвращаем книгу

book1.Return();

// Проверяем снова

Console.WriteLine($"Книга '{book1.Title}' доступна: {book1.IsAvailable()}");

}

}

На рисунке 4 представлен результат выполнения программы.

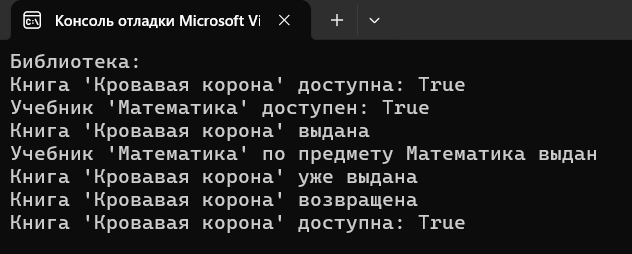


Рисунок 4 – Вывод данных из библиотеки

**Выполнение задания 5.** Создайте приложение для рисования на холсте с использованием интерфейсов. Создайте интерфейс "Рисунок" с методами для рисования линий, кругов и прямоугольников. Реализуйте этот интерфейс в классе для работы с холстом.

Листинг 5. Код программы приложения для рисования на холсте

using System;

// Интерфейс для рисования

interface IDrawing

{

void Draw();

void Clear();

}

// Класс для рисования линий

class LineDrawing : IDrawing

{

public int X1, Y1, X2, Y2;

public void Draw()

{

Console.WriteLine($"Рисую линию от ({X1},{Y1}) до ({X2},{Y2})");

}

public void Clear()

{

Console.WriteLine("Стираю линию");

}

}

// Класс для рисования кругов

class CircleDrawing : IDrawing

{

public int X, Y, Radius;

public void Draw()

{

Console.WriteLine($"Рисую круг в точке ({X},{Y}) с радиусом {Radius}");

}

public void Clear()

{

Console.WriteLine("Стираю круг");

}

}

// Класс для рисования прямоугольников

class RectangleDrawing : IDrawing

{

public int X, Y, Width, Height;

public void Draw()

{

Console.WriteLine($"Рисую прямоугольник в ({X},{Y}) размером {Width}x{Height}");

}

public void Clear()

{

Console.WriteLine("Стираю прямоугольник");

}

}

class Program

{

static void Main()

{

Console.WriteLine("Рисование на холсте:");

// Рисуем линию

LineDrawing line = new LineDrawing();

line.X1 = 10;

line.Y1 = 10;

line.X2 = 50;

line.Y2 = 50;

line.Draw();

// Рисуем круг

CircleDrawing circle = new CircleDrawing();

circle.X = 30;

circle.Y = 30;

circle.Radius = 20;

circle.Draw();

// Рисуем прямоугольник

RectangleDrawing rect = new RectangleDrawing();

rect.X = 5;

rect.Y = 5;

rect.Width = 40;

rect.Height = 30;

rect.Draw();

// Стираем

Console.WriteLine("\nСтираем все:");

line.Clear();

circle.Clear();

rect.Clear();

// Рисуем ещё что-то

Console.WriteLine("\nРисуем новую фигуру:");

LineDrawing line2 = new LineDrawing();

line2.X1 = 0;

line2.Y1 = 0;

line2.X2 = 100;

line2.Y2 = 100;

line2.Draw();

}

}

На рисунке 5 представлен результат выполнения программы.

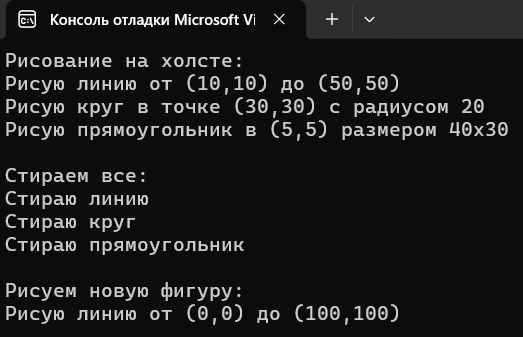


Рисунок 5 – Вывод рисования фигур