**11 Полиморфизм**

Задание 1. Составить программу с одним родительским классом и двумя

потомками. Потомки должны содержать виртуальные функции. Создать виртуальную функцию выдачи результатов расчета методов на экран монитора с указанием названий и полей, и их значений соответствующего объекта. Составить тестирующую программу с выдачей протокола на экран монитора. При этом создать объекты базового и производных типов, используя полиморфный контейнер - массив ссылок базового класса на объекты базового и производных классов.

Листинг программы:

try

{

List<Quad> quads = new List<Quad>() { new Quad("Параллелограмм"), new Quad("Четырёхугольник"), new Rectangle("Прямоугольник",5,4),

new Rectangle("Прямоугольник",5,3), new Square("Квадрат", 4), new Square("Квадрат", 10)};

foreach (Quad item in quads)

{

item.Print();

Console.WriteLine();

}

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

class Quad

{

public string name;

public Quad(string name) => this.name = name;

public virtual void Print()

{

Console.WriteLine($"Название четырёхугольника: {name}");

}

}

class Rectangle : Quad

{

int width;

int length;

public Rectangle(string name, int width, int length) : base(name)

{

this.width = width;

this.length = length;

}

public override void Print()

{

Console.WriteLine($"Название четырхугольника: {name}, плошадь фигуры: {width \* length}");

}

}

class Square : Quad

{

int length;

public Square(string name, int length) : base(name) => this.length = length;

public override void Print()

{

Console.WriteLine($"Название четырхугольника: {name}, плошадь фигуры: {Math.Pow(length, 2)}");

}

}

Анализ результатов:

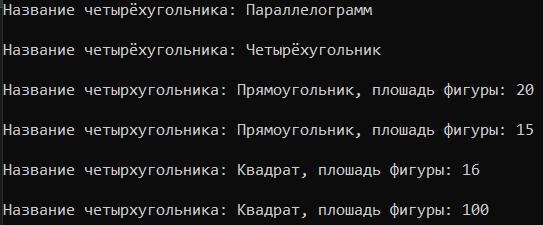


Рисунок 11.1 – Результат работы программы

Источник: собственная разработка

Задание 2. В рамках консольного приложения разработать класс В -наследник класса А (класс А с полями а и b и свойством с. Свойство – значение выражения над полями а и b (выражение и типы полей – см. вариант в таблице 1). Поля инициализировать при объявлении класса. Конструктор оставить по умолчанию. Проследить, чтобы поля а и b напрямую в других классах были недоступны.) с полем d и свойством с2. Свойство с2 – результат вычисления выражения над полями a, b, d. В теле свойства использовать управляющий оператор (см. Вариант в таблице 1) Для класса В определить 2 конструктора: один – наследуется от конструктора класса А, второй –собственный. В теле

программы создать объекты классов А и В, продемонстрировав работу всех конструкторов. Вывести значения свойства на экран.

Листинг программы:

try

{

List<Quad> quads = new List<Quad>() {new Rectangle("Прямоугольник",5,4),

new Rectangle("Прямоугольник",5,3), new Square("Квадрат", 4), new Square("Квадрат", 10)};

int max = 0;

foreach (Quad item in quads)

{

item.Print();

if (item.Sqr > max)

max = item.Sqr;

Console.WriteLine();

}

Console.WriteLine($"Максимальная площадь: {max}");

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

abstract class Quad

{

public abstract string Name { get; set; }

public abstract int Sqr { get; set; }

public abstract void Print();

}

class Rectangle : Quad

{

int width;

int length;

public override int Sqr { get; set; }

public override string Name { get; set; }

public Rectangle(string name, int width, int length)

{

this.width = width;

this.length = length;

this.Name = name;

Sqr = width \* length;

}

public override void Print()

{

Console.WriteLine($"Название четырхугольника: {Name}, плошадь фигуры: {Sqr}");

}

}

class Square : Quad

{

int length;

public override int Sqr { get; set; }

public override string Name { get; set; }

public Square(string name, int length)

{

this.Name = name;

this.length = length;

Sqr = (int)Math.Pow(length, 2);

}

public override void Print()

{

Console.WriteLine($"Название четырхугольника: {Name}, плошадь фигуры: {Sqr}");

}

Анализ результатов:

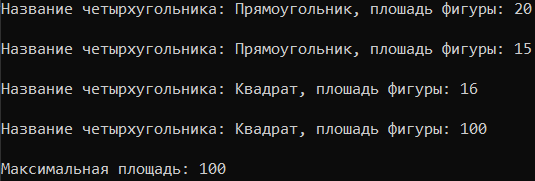


Рисунок 11.2 – Результат работы программы

Источник: собственная разработка