|  |
| --- |
| Министерство образования Республики Беларусь  УО «Полоцкий государственный университет им. Ефросинии Полоцкой» |

|  |  |
| --- | --- |
|  | Факультет информационных технологий  Кафедра технологий программирования |

|  |
| --- |
| ОТЧЕТ |
| **к лабораторной работе №8(4)**  **Дисциплина**: «Программирование на платформе .NET» |

|  |  |
| --- | --- |
| Выполнил: | Студент группы 19-ИТ-2  Гилимович А.Н. |

|  |  |
| --- | --- |
| Проверила: | Преподователь  Виноградова А.Д. |

|  |
| --- |
| Полоцк, 2022 г. |

**Задание**

1. Программа должна быть разработана в виде консольного приложения на языке F#.

2. Абстрактный класс «Геометрическая фигура» содержит виртуальный метод для вычисления площади фигуры.

3. Класс «Прямоугольник» наследуется от класса «Геометрическая фигура». Ширина и высота объявляются как свойства (property). Класс должен содержать конструктор по параметрам «ширина» и «высота».

4. Класс «Квадрат» наследуется от класса «Прямоугольник». Класс должен содержать конструктор по длине стороны.

5. Класс «Круг» наследуется от класса «Геометрическая фигура». Радиус объявляется как свойство (property). Класс должен содержать конструктор по параметру «радиус».

6. Для классов «Прямоугольник», «Квадрат», «Круг» переопределить виртуальный метод Object.ToString(), который возвращает в виде строки основные параметры фигуры и ее площадь.

7. Разработать интерфейс IPrint. Интерфейс содержит метод Print(), который не принимает параметров и возвращает void. Для классов «Прямоугольник», «Квадрат», «Круг» реализовать наследование от интерфейса IPrint. Переопределяемый метод Print() выводит на консоль информацию, возвращаемую переопределенным методом ToString().

8. Реализуйте класс геометрическая фигура в виде алгебраического типа (discriminated union), который содержит варианты (дискриминаторы) «Прямоугольник», «Квадрат», «Круг» с необходимыми параметрами.

9. Разработайте для данного класса функцию вычисления площади. Функция должна принимать параметр типа «геометрическая фигура» и вычислять различные варианты площади в зависимости от дискриминатора. Необходимо использовать механизм сопоставления с образцом. (Используйте пример вычисления корней квадратного уравнения)

**Ход работы**

Изучив теоритический материал, преступил к выполнению заданий.

**Листинг 1** – Абстрактный класс

[<AbstractClass>]

type GiometrycFigure() =

abstract member setArea: unit -> float

**Листинг 2** – Интерфейс

type Interface1 = interface

abstract member Print: unit -> unit

end

**Листинг 3** – класс Rectangle

type Rectangle(a: float, b: float) =

inherit GiometrycFigure()

let mutable A = a

let mutable B = b

override this.setArea() = A \* B

override this.ToString() =

let str = "\nПрямоугольник \na: " + A.ToString() + "\nb: " + B.ToString() + "\nПлощадь прямоугольника: " + (this.setArea()).ToString()

str

interface Interface1 with

member this.Print() =

printf "%s" (this.ToString())

**Листинг 4** – кдасс Square

type Square(a: float) =

inherit GiometrycFigure()

let mutable A = a

override this.setArea() = A \* A

override this.ToString() =

let str = "\n\nКвадрат\na: " + A.ToString() + "\nПлощадь квадрата: " + (this.setArea()).ToString()

str

interface Interface1 with

member this.Print() =

printf "%s" (this.ToString())

**Листинг 5** – кдасс Circle

type Circle(a: float) =

inherit GiometrycFigure()

let mutable A = a

override this.setArea() = A \* A \* 3.14

override this.ToString() = "\n\nКруг\nr: " + A.ToString() + "\nПлощадь круга: " + (this.setArea()).ToString()

interface Interface1 with

member this.Print() =

printf "%s" (this.ToString())

**Листинг 6** – тестирование первой части

let rectangle = Rectangle(1.2, 10)

let i1 = rectangle :> Interface1

i1.Print()

let square = Square(5)

let i2 = square :> Interface1

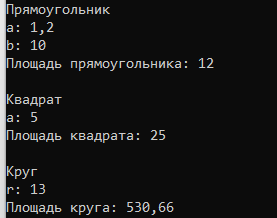
i2.Print()

let circle = Circle(13)

let i3 = circle :> Interface1

i3.Print()

Результат тестирования первой части представленн на рисунке 1



**Рисунок 1 –** тестирование первой части

**Листинг 7** – класс в виде алгебраического типа

type GFigure =

| Rectangle1 of a: float \* b : float

| Square1 of a: float

| Circle1 of r: float

let gArea (gf: GFigure) =

match gf with

|GFigure.Rectangle1 (a, b) -> printfn"\nСтороны прямоугольника: %s %s\nПлощадь прямоугольника: %s" (a.ToString()) (b.ToString()) ((a\*b).ToString())

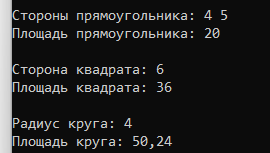
|GFigure.Square1 a -> printfn"\nСторона квадрата: %s \nПлощадь квадрата: %s" (a.ToString()) ((a\*a).ToString())

|GFigure.Circle1 r -> printfn"\nРадиус круга: %s \nПлощадь круга: %s" (r.ToString()) ((r\*r\*3.14).ToString())

gArea (GFigure.Rectangle1(4, 5))

gArea (GFigure.Square1 6)

gArea (GFigure.Circle1 4)



**Рисунок 2 –** тестирование второй части программы

**Вывод:** изучили основы F# использования классов, интерфейсов и наследования. Научились работать с алгебраическим типом и дескрипторами. Выполнили задания лабараторной работы и усвоили теоретический материал.