Студент: Лачина Екатерина
Группа: ИУ5-33Б

Отчет по лабораторной работе №1

«Основные конструкции языка Python (функциональная парадигма, ООП парадигма) и Golang»

1) **Цель лабораторной работы:** изучение основных конструкций языка Python.

2) **Условие задачи:**

Разработать программу для решения биквадратного уравнения.

1. Программа должна быть разработана в виде консольного приложения на языке Python.
2. Программа осуществляет ввод с клавиатуры коэффициентов A, B, C, вычисляет дискриминант и ДЕЙСТВИТЕЛЬНЫЕ корни уравнения (в зависимости от дискриминанта).
3. Коэффициенты A, B, C могут быть заданы в виде параметров командной строки (вариант задания параметров приведен в конце файла с примером кода). Если они не заданы, то вводятся с клавиатуры в соответствии с пунктом 2. Описание работы с параметрами командной строки.
4. Если коэффициент A, B, C введен или задан в командной строке некорректно, то необходимо проигнорировать некорректное значение и вводить коэффициент повторно пока коэффициент не будет введен корректно. Корректно заданный коэффициент - это коэффициент, значение которого может быть без ошибок преобразовано в действительное число.
5. Дополнительное задание 1 (*). Разработайте две программы на языке Python - одну с применением процедурной парадигмы, а другую с применением объектно-ориентированной парадигмы.

6. Дополнительное задание 2 (*). Разработайте две программы - одну на языке Python, а другую на любом другом языке программирования (кроме C++).

3) Реализация задачи на языке программирования python с использованием процедурной парадигмы:

```
import math

def check(a, b, c):
    if a == 0 and b == 0 and c == 0:
        print("x принадлежит R")
        return True
    elif a == 0 and c == 0:
        print("x = 0")
        return True
    elif a == 0 and b == 0:
        print("Нет действительных корней")
        return True
    elif a == 0 and b != 0 and c != 0:
        # bx^2 = -c
        # x^2 = -c/b
        t = -c/b
        if t > 0:
            x1 = math.sqrt(t)
            x2 = -math.sqrt(t)
            res = [x1, x2]
        elif t == 0:
            x = math.sqrt(t)
            res = [x]
        else:
            print("Нет действительных корней")
        print("Корни уравнения: ", *res)
        return True
    return False

# at^2+bt+c=0
def discriminant(a, b, c):
    D = b**2 - 4*a*c
    return D

def Solution_sqrt_equation(a, b, c):
    D = discriminant(a, b, c)
    if D > 0:
        t1 = (-b + math.sqrt(D))/(2*a)
        t2 = (-b - math.sqrt(D)) / (2 * a)
        return [t1, t2]
    elif D == 0:
        t = (-b)/(2*a)
        return [t]
    else:
        return []

def Rrturn_X(a, b, c):
    T_res = Solution_sqrt_equation(a, b, c)
    res = []
    for t in T_res:
        if t > 0:
            res.append(math.sqrt(t))
            res.append(-math.sqrt(t))
        elif t == 0:
```

```
            res.append(0)
    return res

def check_abc(prompt):
    while True:
        try:
            return float(input(prompt))
        except ValueError:
            print("Некорректный ввод. Введите число")

def main():
    a = check_abc("Введите коэффициент A: ")
    b = check_abc("Введите коэффициент B: ")
    c = check_abc("Введите коэффициент C: ")

    if check(a, b, c):
        return

    X_res = Rrturn_X(a, b, c)
    if X_res:
        print("Корни уравнения: ", *X_res)
    else:
        print("Нет действительных корней")

if __name__ == "__main__":
    main()
```

3.1) Экранные формы с примерами выполнения программы:

```
Введите коэффициент A: 1
Введите коэффициент B: 1
Введите коэффициент C: 5
Нет действительных корней
```

```
Введите коэффициент A: 0
Введите коэффициент B: 0
Введите коэффициент C: 0
x принадлежит R
```

```
Введите коэффициент A: 2
Введите коэффициент B: -6
Введите коэффициент C: 4
Корни уравнения:  1.4142135623730951 -1.4142135623730951 1.0 -1.0

Process finished with exit code 0
```

```
Введите коэффициент A: 3
Введите коэффициент B: -5
Введите коэффициент C: 1
Корни уравнения:  1.1976053381271583 -1.1976053381271583 0.48208725429739563 -0.48208725429739563

Process finished with exit code 0
```

```
Введите коэффициент A: вгррмоав
Некорректный ввод. Введите число
Введите коэффициент A: 3
Введите коэффициент B: 9
Введите коэффициент C: 0
Корни уравнения:  0

Process finished with exit code 0
```

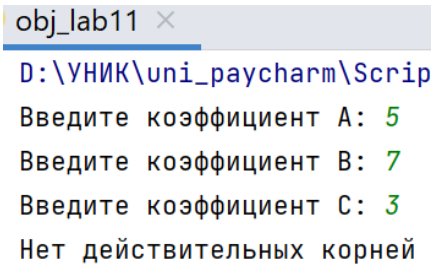
```
Введите коэффициент A: 1
Введите коэффициент B: 0
Введите коэффициент C: 4
Нет действительных корней
```

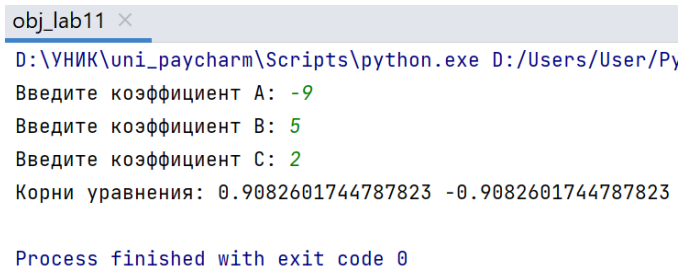
4) Реализация задачи на языке программирования python с использованием ООП парадигмы:

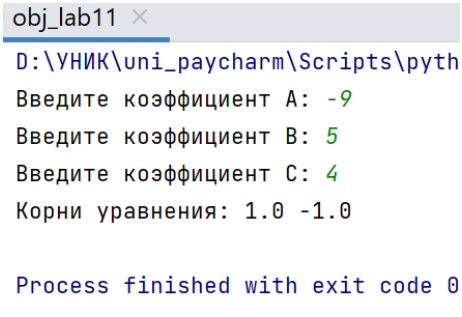
```python
import math

class BiQuadraticEquation:
    def __init__(self, a, b, c):
        self.a = a
        self.b = b
        self.c = c

    def check(self):
        if self.a == 0 and self.b == 0 and self.c == 0:
            print("x принадлежит R")
            return True
        elif self.a == 0 and self.c == 0:
            print("x = 0")
            return True
        elif self.a == 0 and self.b == 0:
            print("Нет действительных корней")
            return True
        elif self.a == 0 and self.b != 0 and self.c != 0:
            t = -self.c / self.b
            if t > 0:
                x1 = math.sqrt(t)
                x2 = -math.sqrt(t)
                res = [x1, x2]
            elif t == 0:
                x = math.sqrt(t)
                res = [x]
            else:
                print("Нет действительных корней")
            print("Корни уравнения: ", *res)
            return True
        return False

    def discriminant(self):
        return self.b**2 - 4 * self.a * self.c

    def Solution_sqrt_equation(self):
        D = self.discriminant()
        if D > 0:
            t1 = (-self.b + math.sqrt(D)) / (2 * self.a)
            t2 = (-self.b - math.sqrt(D)) / (2 * self.a)
            return [t1, t2]
        elif D == 0:
            t = -self.b / (2 * self.a)
            return [t]
        else:
```

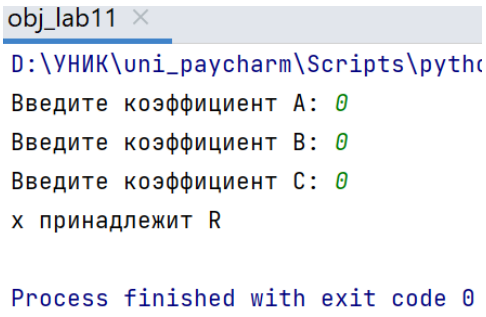
```
            return []

    def Rrturn_X(self):
        T_res = self.Solution_sqrt_equation()
        res = []
        for t in T_res:
            if t > 0:
                res.append(math.sqrt(t))
                res.append(-math.sqrt(t))
            elif t == 0:
                res.append(0)
        return res

    def solve(self):
        if self.check():
            return
        x_res = self.Rrturn_X()
        if x_res:
            print("Корни уравнения:", *x_res)
        else:
            print("Нет действительных корней")


class InputHandler:
    def check_abc(prompt):
        while True:
            try:
                return float(input(prompt))
            except ValueError:
                print("Некорректный ввод. Введите число")


def main():
    a = InputHandler.check_abc("Введите коэффициент A: ")
    b = InputHandler.check_abc("Введите коэффициент B: ")
    c = InputHandler.check_abc("Введите коэффициент C: ")
    equation = BiQuadraticEquation(a, b, c)
    equation.solve()


if __name__ == "__main__":
    main()
```

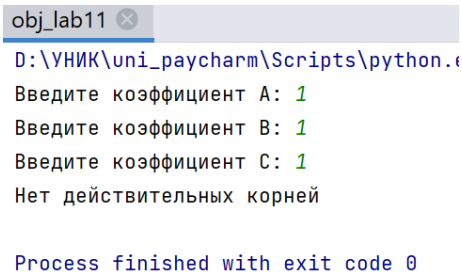
4.1) Экранные формы с примерами выполнения программы:

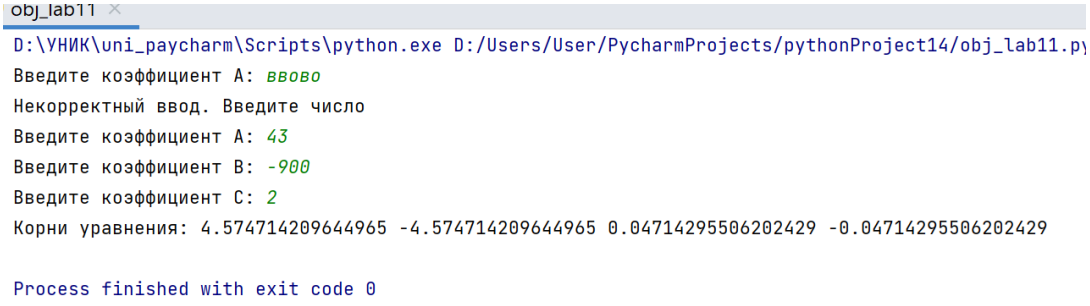












5) Реализация задачи на языке программирования golang с функциональной парадигмы:

```
package main

import (
    "fmt"
    "math"
)

func check(a, b, c float64) bool {
    if a == 0 && b == 0 && c == 0 {
        fmt.Println("x принадлежит R")
        return true
    } else if a == 0 && c == 0 {
        fmt.Println("x = 0")
        return true
    } else if a == 0 && b == 0 {
        fmt.Println("Нет действительных корней")
        return true
    } else if a == 0 && b != 0 && c != 0 {
```

```
        t := -c / b
        if t > 0 {
            x1 := math.Sqrt(t)
            x2 := -math.Sqrt(t)
            fmt.Printf("Корни уравнения: %g, %g\n", x1, x2)
        } else if t == 0 {
            fmt.Println("x = 0")
        } else {
            fmt.Println("Нет действительных корней")
        }
        return true
    }
    return false
}

func discriminant(a, b, c float64) float64 {
    return b*b - 4*a*c
}

func solveQuadratic(a, b, c float64) []float64 {
    D := discriminant(a, b, c)
    if D > 0 {
        t1 := (-b + math.Sqrt(D)) / (2 * a)
        t2 := (-b - math.Sqrt(D)) / (2 * a)
        return []float64{t1, t2}
    } else if D == 0 {
        t := -b / (2 * a)
        return []float64{t}
    }
    return nil
}

func returnX(a, b, c float64) []float64 {
    tRes := solveQuadratic(a, b, c)
    var res []float64

    for _, t := range tRes {
        if t > 0 {
            res = append(res, math.Sqrt(t), -math.Sqrt(t))
        } else if t == 0 {
            res = append(res, 0)
        }
```

```
    }
    return res
}

func checkABC(prompt string) float64 {
    var value float64
    for {
        fmt.Print(prompt)
        _, err := fmt.Scan(&value)
        if err == nil {
            break
        } else {
            fmt.Println("Некорректный ввод. Введите число.")
        }
    }
    return value
}

func main() {
    a := checkABC("Введите коэффициент A: ")
    b := checkABC("Введите коэффициент B: ")
    c := checkABC("Введите коэффициент C: ")

    if check(a, b, c) {
        return
    }

    xRes := returnX(a, b, c)
    if len(xRes) > 0 {
        fmt.Print("Корни уравнения: ")
        for _, x := range xRes {
            fmt.Printf("%g ", x)
        }
        fmt.Println()
    } else {
        fmt.Println("Нет действительных корней")
    }
}
```

5.1) Экранные формы с примерами выполнения программы:

```
PS D:\ro> cd univer
PS D:\ro\univer> go run lab1.go
Введите коэффициент A: -9
Введите коэффициент B: 5
Введите коэффициент C: 4
Корни уравнения: 1 -1
PS D:\ro\univer> go run lab1.go
Введите коэффициент A: 1
Введите коэффициент B: 1
Введите коэффициент C: 5
Нет действительных корней
PS D:\ro\univer> go run lab1.go
Введите коэффициент A: 0
Введите коэффициент B: 0
Введите коэффициент C: 0
x принадлежит R
PS D:\ro\univer> go run lab1.go
Введите коэффициент A: 2
Введите коэффициент B: -6
Введите коэффициент C: 4
Корни уравнения: 1.4142135623730951 -1.4142135623730951 1 -1
PS D:\ro\univer> go run lab1.go
Введите коэффициент A: 3
Введите коэффициент B: -5
Введите коэффициент C: 1
Корни уравнения: 1.1976053381271583 -1.1976053381271583 0.48208725429739563 -0.48208725429739563

PS D:\ro\univer> go run lab1.go
Введите коэффициент A: 1
Введите коэффициент B: 0
Введите коэффициент C: 4
Нет действительных корней

PS D:\ro\univer> go run lab1.go
Введите коэффициент A: 43
Введите коэффициент B: -900
Введите коэффициент C: 2
Корни уравнения: 4.574714209644965 -4.574714209644965 0.04714295506202429 -0.04714295506202429
PS D:\ro\univer>
```