

Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
Московский государственный технический университет имени Н.Э.Баумана

Отчет
по домашнему заданию
по дисциплине «Базовые компоненты интернет-технологий»

Выполнила студентка группы ИУ5-34
Изьюрова Маргарита

Проверил
Гапанюк Юрий Евгеньевич

Москва, 2017 год

На основе рассмотренного примера составить программу на функциональном языке программирования для решения биквадратного уравнения с использованием алгоритма рассмотренного в разделе «Биквадратное уравнение» статьи https://ru.wikipedia.org/wiki/Уравнение_четвёртой_степени. Программа должна использовать алгебраические типы и механизм сопоставления с образцом.

В случае комплексных корней их вычисление не обязательно, можно выводить информацию о том, что корни комплексные.

Код программы:

```
open System
type BiquadraticRootEmpty = interface end
type NoRoots() =
    interface BiquadraticRootEmpty
type OneRoot(p: double) =
    interface BiquadraticRootEmpty
    member val root = p: double with get, set
type TwoRoots(p1:double, p2:double) =
    interface BiquadraticRootEmpty
    member val root1 = p1: double with get, set
    member val root2 = p2: double with get, set
type ThreeRoots(p1: double, p2: double, p3:double) =
    interface BiquadraticRootEmpty
    member val root1 = p1: double with get, set
    member val root2 = p2: double with get, set
    member val root3 = p3: double with get, set
type FourRoots(p1:double, p2:double, p3:double, p4:double) =
    interface BiquadraticRootEmpty
    member val root1 = p1: double with get, set
    member val root2 = p2: double with get, set
    member val root3 = p3: double with get, set
    member val root4 = p4: double with get, set
let CalculateRoots(a:double, b:double, c:double): BiquadraticRootEmpty =
    let D = b*b - 4.0*a*c;
    if (D = 0.0) then
        let x = -b/(2.0*a)
        if (x < 0.0) then (NoRoots() :> BiquadraticRootEmpty)
        else if (x = 0.0) then (OneRoot(Math.Sqrt(x)) :> BiquadraticRootEmpty)
        else (TwoRoots(-Math.Sqrt(x), Math.Sqrt(x)) :> BiquadraticRootEmpty)
    else if (D > 0.0) then
        let sqrtD = Math.Sqrt(D)
        let x1 = (-b-sqrtD)/(2.0*a);
        let x2 = (-b+sqrtD)/(2.0*a);
        if (x1 > 0.0 && x2 > 0.0) then (FourRoots(-Math.Sqrt(x1),-
Math.Sqrt(x2),Math.Sqrt(x1),Math.Sqrt(x2)) :> BiquadraticRootEmpty)
```

```

    else if (x1 > 0.0 && x2 = 0.0) then (ThreeRoots(-
Math.Sqrt(x1),x2,Math.Sqrt(x1)) :> BiquadraticRootEmpty)
    else if (x1 > 0.0 && x2 < 0.0) then (TwoRoots(-
Math.Sqrt(x1),Math.Sqrt(x1)) :> BiquadraticRootEmpty)
    else if (x1 = 0.0 && x2 > 0.0) then (ThreeRoots(-
Math.Sqrt(x2),x1,Math.Sqrt(x2)) :> BiquadraticRootEmpty)
    else if (x1 = 0.0 && x2 < 0.0) then (OneRoot(x1) :> BiquadraticRootEmpty)
    else if (x1 < 0.0 && x2 > 0.0) then (TwoRoots(-
Math.Sqrt(x2),Math.Sqrt(x2)) :> BiquadraticRootEmpty)
    else if (x1 < 0.0 && x2 = 0.0) then (OneRoot(x2) :> BiquadraticRootEmpty)
    else if (x1 = 0.0 && x2 = 0.0) then (OneRoot(x1) :> BiquadraticRootEmpty)
    else (NoRoots() :> BiquadraticRootEmpty)
else
    (NoRoots() :> BiquadraticRootEmpty)
let PrintRoots(a: double, b: double, c: double):unit =
    printfn "Коэффициенты: a = %A, b = %A, c = %A" a b c
    let root = CalculateRoots(a, b, c)
    let textResult =
        match root with
        | :? NoRoots -> "Корней нет"
        | :? OneRoot as r -> "Один корень: " + r.root.ToString()
        | :? TwoRoots as r -> "Два корня: " + r.root1.ToString() + ", " + r.root2.ToString()
        | :? ThreeRoots as r -
> "Три корня: " + r.root1.ToString() + ", " + r.root2.ToString() + ", " + r.root3.ToString()
        | :? FourRoots as r -
> "Четыре корня: " + r.root1.ToString() + ", " + r.root2.ToString() + ", " + r.root3.ToString()
        + ", " + r.root4.ToString()
        | _ -> ""
    printfn "%s" textResult
[<EntryPoint>]
let main argv =
    let a1 = 1.0;
    let b1 = -2.0;
    let c1 = 1.0;
    PrintRoots(a1, b1, c1);
    let a2 = 3.0;
    let b2 = -25.0;
    let c2 = 5.0;
    PrintRoots(a2, b2, c2);
    let a3 = 1.0;
    let b3 = 25.0;
    let c3 = -1.0;
    PrintRoots(a3, b3, c3);
    0 // return an integer exit code

```

Примеры выполнения программы:

Коэффициенты: $a = 1.0$, $b = -2.0$, $c = 1.0$

Два корня: -1 , 1

Коэффициенты: $a = 3.0$, $b = -25.0$, $c = 5.0$

Четыре корня: $-0,452819197176712$, $-2,85101527670435$, $0,452819197176712$, $2,85101527670435$

Коэффициенты: $a = 1.0$, $b = 25.0$, $c = -1.0$

Два корня: $-0,199840446317689$, $0,199840446317689$