

Міністерство освіти і науки України
Національний технічний університет України «Київський
політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки

Кафедра ІІІ

Звіт

з лабораторної роботи № 4 з дисципліни

«Основи програмування 2»

„УСПАДКУВАННЯ ТА ПОЛІМОРФІЗМ”

Виконав

ІІІ-15 Костін Вадим
(шифр, прізвище, ім'я, по батькові)

Перевірів(ла)

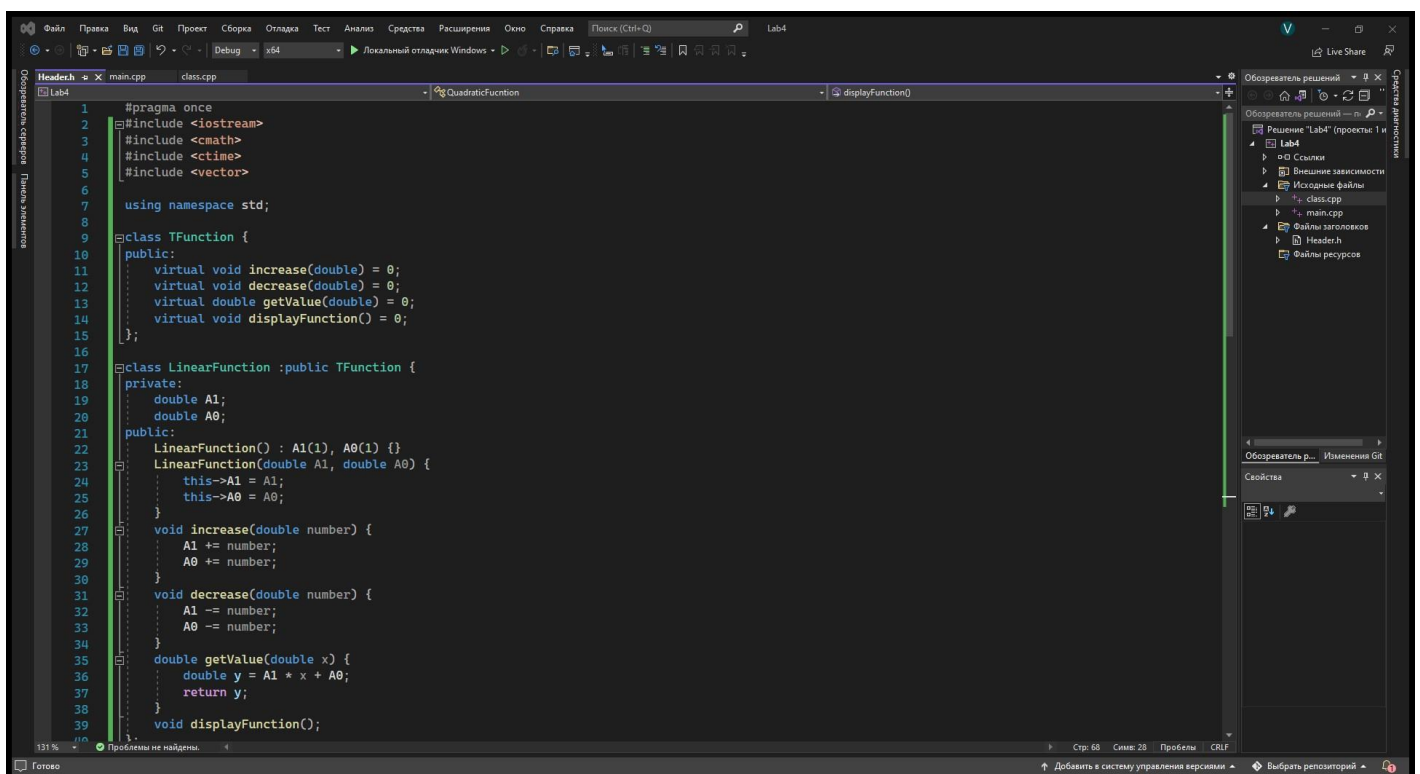
Вєчерковська Анастасія Сергіївна
(прізвище, ім'я, по батькові)

Київ 2022

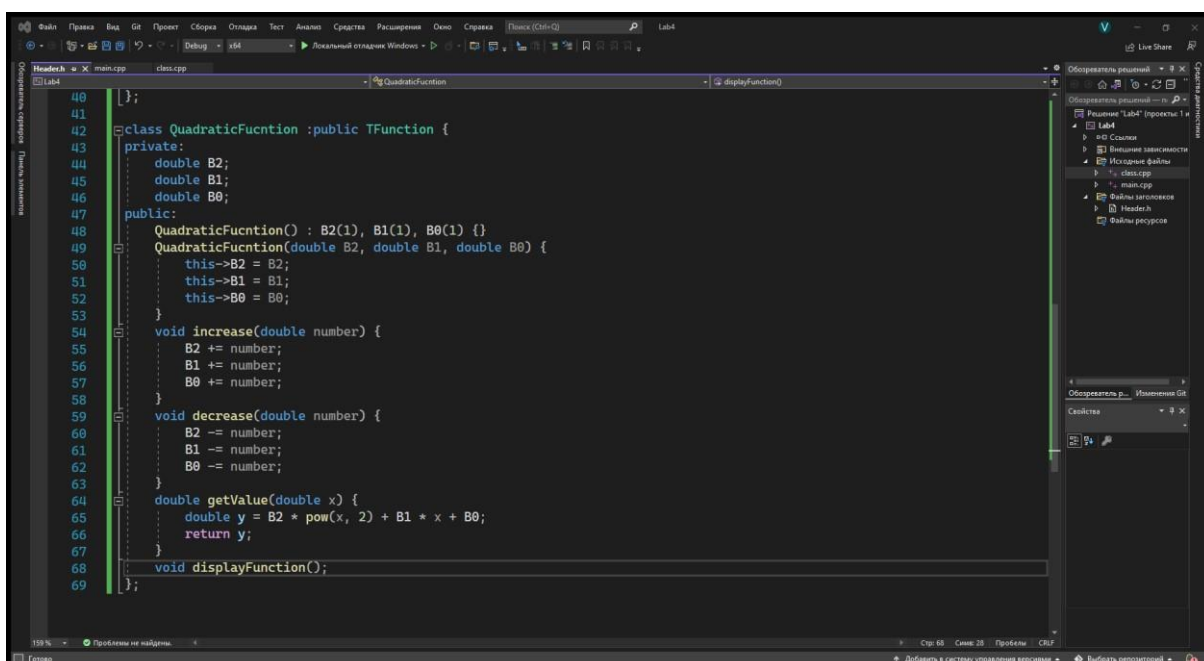
Завдання

15. Спроекувати клас TFunction, який представляє функцію і містить методи збільшення / зменшення всіх коефіцієнтів функції на вказану величину та обчислення значення функції в заданій точці. На основі цього класу створити класи-нащадки “Лінійна функція” (виду $a_1x + a_0$) та “Квадратична функція” (виду $b_2x^2 + b_1x + b_0$). Створити n лінійних функцій та m квадратичних функцій, згенерувавши дані для них випадковим чином. Коефіцієнти лінійних функцій збільшити на 3, а квадратичних - зменшити на 2. Визначити функцію, яка має найбільше значення у введеній користувачем точці.

Програма на C++ Header

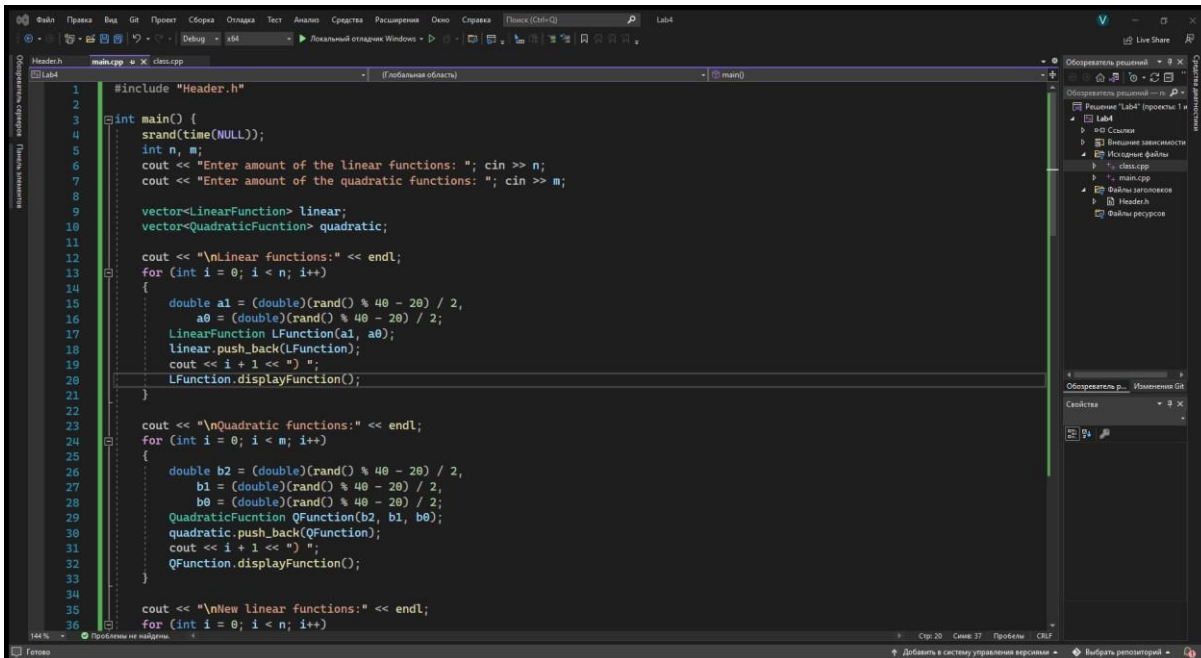


```
1 #pragma once
2 #include <iostream>
3 #include <cmath>
4 #include <ctime>
5 #include <vector>
6
7 using namespace std;
8
9 class TFunction {
10 public:
11     virtual void increase(double) = 0;
12     virtual void decrease(double) = 0;
13     virtual double getValue(double) = 0;
14     virtual void displayFunction() = 0;
15 };
16
17 class LinearFunction : public TFunction {
18 private:
19     double A1;
20     double A0;
21 public:
22     LinearFunction() : A1(1), A0(1) {}
23     LinearFunction(double A1, double A0) {
24         this->A1 = A1;
25         this->A0 = A0;
26     }
27     void increase(double number) {
28         A1 += number;
29         A0 += number;
30     }
31     void decrease(double number) {
32         A1 -= number;
33         A0 -= number;
34     }
35     double getValue(double x) {
36         double y = A1 * x + A0;
37         return y;
38     }
39     void displayFunction();
40 }
```



```
40 };
41
42 class QuadraticFunction : public TFunction {
43 private:
44     double B2;
45     double B1;
46     double B0;
47 public:
48     QuadraticFunction() : B2(1), B1(1), B0(1) {}
49     QuadraticFunction(double B2, double B1, double B0) {
50         this->B2 = B2;
51         this->B1 = B1;
52         this->B0 = B0;
53     }
54     void increase(double number) {
55         B2 += number;
56         B1 += number;
57         B0 += number;
58     }
59     void decrease(double number) {
60         B2 -= number;
61         B1 -= number;
62         B0 -= number;
63     }
64     double getValue(double x) {
65         double y = B2 * pow(x, 2) + B1 * x + B0;
66         return y;
67     }
68     void displayFunction();
69 }
```

Main



```
#include "Header.h"

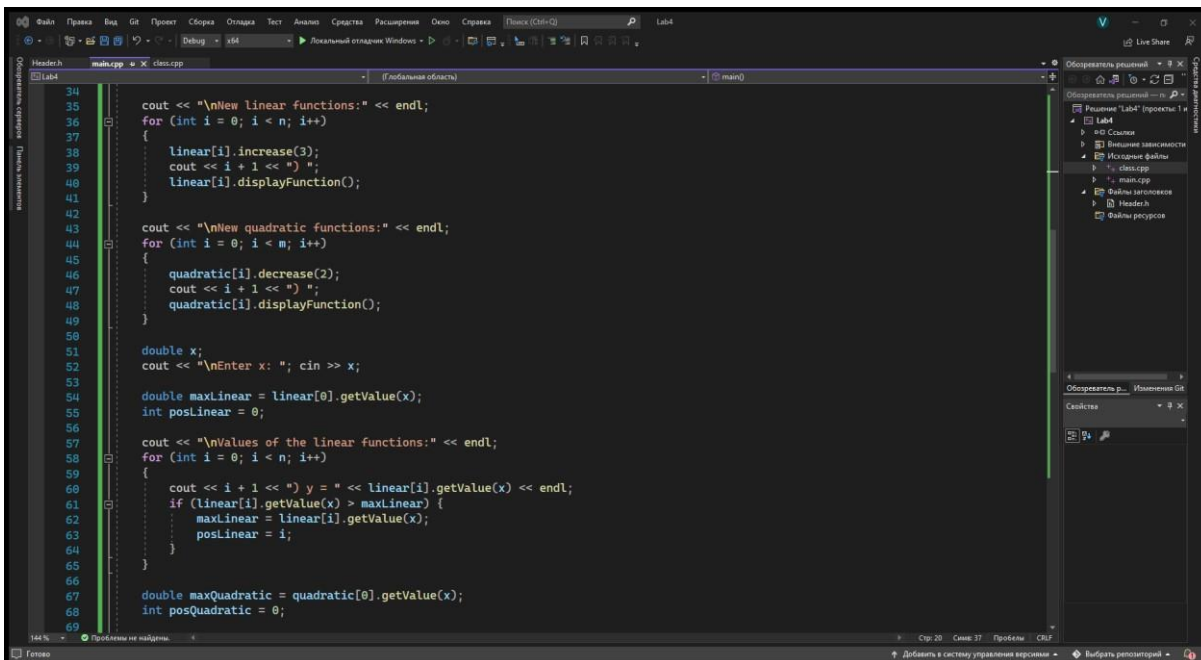
int main() {
    srand(time(NULL));
    int n, m;
    cout << "Enter amount of the linear functions: "; cin >> n;
    cout << "Enter amount of the quadratic functions: "; cin >> m;

    vector<LinearFunction> linear;
    vector<QuadraticFunction> quadratic;

    cout << "\nLinear functions:" << endl;
    for (int i = 0; i < n; i++)
    {
        double a1 = (double)(rand() % 40 - 20) / 2;
        double a0 = (double)(rand() % 40 - 20) / 2;
        LinearFunction LFunction(a1, a0);
        linear.push_back(LFunction);
        cout << i + 1 << " ";
        LFunction.displayFunction();
    }

    cout << "\nQuadratic functions:" << endl;
    for (int i = 0; i < m; i++)
    {
        double b2 = (double)(rand() % 40 - 20) / 2;
        double b1 = (double)(rand() % 40 - 20) / 2;
        double b0 = (double)(rand() % 40 - 20) / 2;
        QuadraticFunction QFunction(b2, b1, b0);
        quadratic.push_back(QFunction);
        cout << i + 1 << " ";
        QFunction.displayFunction();
    }

    cout << "\nNew linear functions:" << endl;
    for (int i = 0; i < n; i++)
```



```
    cout << "\nNew linear functions:" << endl;
    for (int i = 0; i < n; i++)
    {
        linear[i].increase(3);
        cout << i + 1 << " ";
        linear[i].displayFunction();
    }

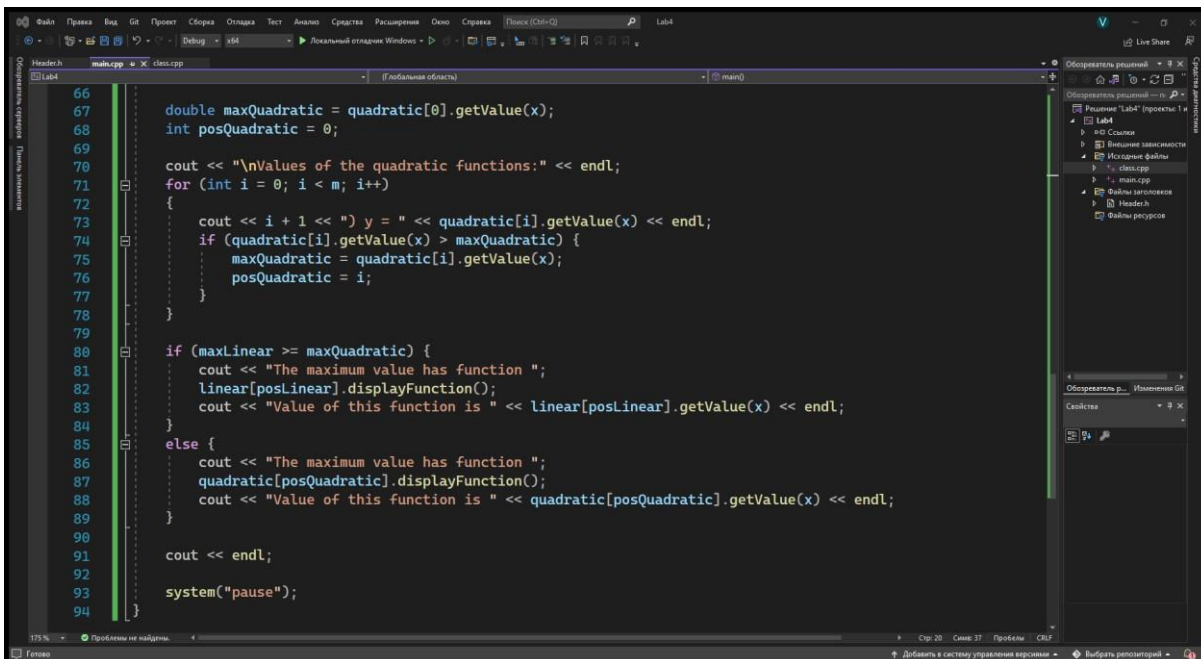
    cout << "\nNew quadratic functions:" << endl;
    for (int i = 0; i < m; i++)
    {
        quadratic[i].decrease(2);
        cout << i + 1 << " ";
        quadratic[i].displayFunction();
    }

    double x;
    cout << "\nEnter x: "; cin >> x;

    double maxLinear = linear[0].getValue(x);
    int posLinear = 0;

    cout << "\nValues of the linear functions:" << endl;
    for (int i = 0; i < n; i++)
    {
        cout << i + 1 << " y = " << linear[i].getValue(x) << endl;
        if (linear[i].getValue(x) > maxLinear) {
            maxLinear = linear[i].getValue(x);
            posLinear = i;
        }
    }

    double maxQuadratic = quadratic[0].getValue(x);
    int posQuadratic = 0;
```



```
    double maxQuadratic = quadratic[0].getValue(x);
    int posQuadratic = 0;

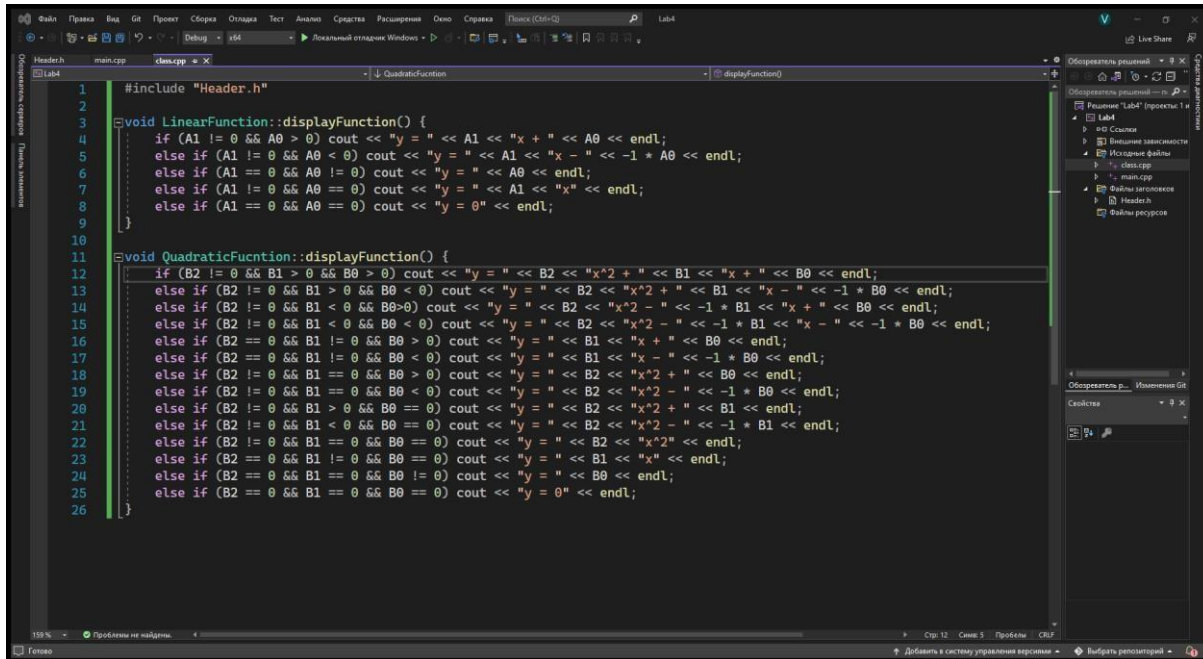
    cout << "\nValues of the quadratic functions:" << endl;
    for (int i = 0; i < m; i++)
    {
        cout << i + 1 << " y = " << quadratic[i].getValue(x) << endl;
        if (quadratic[i].getValue(x) > maxQuadratic) {
            maxQuadratic = quadratic[i].getValue(x);
            posQuadratic = i;
        }
    }

    if (maxLinear >= maxQuadratic) {
        cout << "The maximum value has function ";
        linear[posLinear].displayFunction();
        cout << "Value of this function is " << linear[posLinear].getValue(x) << endl;
    }
    else {
        cout << "The maximum value has function ";
        quadratic[posQuadratic].displayFunction();
        cout << "Value of this function is " << quadratic[posQuadratic].getValue(x) << endl;
    }

    cout << endl;

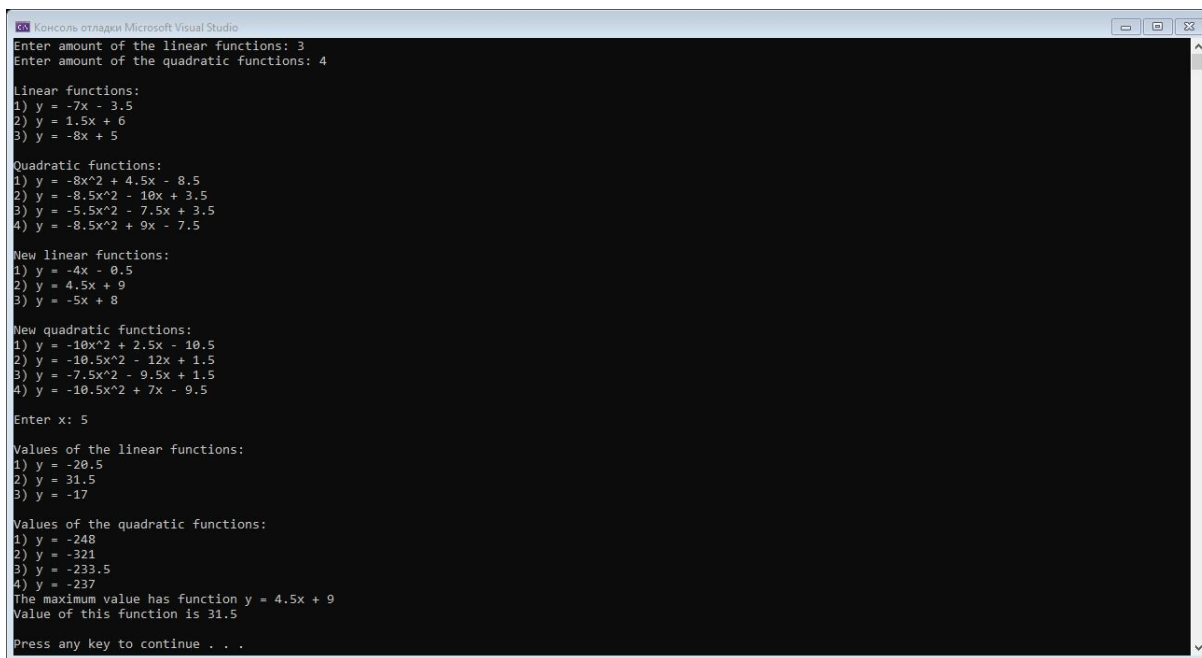
    system("pause");
}
```

Class



```
1 #include "Header.h"
2
3 void LinearFunction::displayFunction() {
4     if (A1 != 0 && A0 > 0) cout << "y = " << A1 << "x + " << A0 << endl;
5     else if (A1 != 0 && A0 < 0) cout << "y = " << A1 << "x - " << -1 * A0 << endl;
6     else if (A1 == 0 && A0 != 0) cout << "y = " << A0 << endl;
7     else if (A1 != 0 && A0 == 0) cout << "y = " << A1 << "x" << endl;
8     else if (A1 == 0 && A0 == 0) cout << "y = 0" << endl;
9 }
10
11 void QuadraticFunction::displayFunction() {
12     if (B2 != 0 && B1 > 0 && B0 > 0) cout << "y = " << B2 << "x^2 + " << B1 << "x + " << B0 << endl;
13     else if (B2 != 0 && B1 > 0 && B0 < 0) cout << "y = " << B2 << "x^2 + " << B1 << "x - " << -1 * B0 << endl;
14     else if (B2 != 0 && B1 < 0 && B0 > 0) cout << "y = " << B2 << "x^2 - " << -1 * B1 << "x + " << B0 << endl;
15     else if (B2 != 0 && B1 < 0 && B0 < 0) cout << "y = " << B2 << "x^2 - " << -1 * B1 << "x - " << -1 * B0 << endl;
16     else if (B2 == 0 && B1 != 0 && B0 > 0) cout << "y = " << B1 << "x + " << B0 << endl;
17     else if (B2 == 0 && B1 != 0 && B0 < 0) cout << "y = " << B1 << "x - " << -1 * B0 << endl;
18     else if (B2 != 0 && B1 == 0 && B0 > 0) cout << "y = " << B2 << "x^2 + " << B0 << endl;
19     else if (B2 != 0 && B1 == 0 && B0 < 0) cout << "y = " << B2 << "x^2 - " << -1 * B0 << endl;
20     else if (B2 != 0 && B1 > 0 && B0 == 0) cout << "y = " << B2 << "x^2 + " << B1 << endl;
21     else if (B2 != 0 && B1 < 0 && B0 == 0) cout << "y = " << B2 << "x^2 - " << -1 * B1 << endl;
22     else if (B2 == 0 && B1 == 0 && B0 > 0) cout << "y = " << B0 << endl;
23     else if (B2 == 0 && B1 != 0 && B0 == 0) cout << "y = " << B1 << "x" << endl;
24     else if (B2 == 0 && B1 == 0 && B0 != 0) cout << "y = " << B0 << endl;
25     else if (B2 == 0 && B1 == 0 && B0 == 0) cout << "y = 0" << endl;
26 }
```

Тестування програми



```
Консоль отладки Microsoft Visual Studio
Enter amount of the linear functions: 3
Enter amount of the quadratic functions: 4

Linear functions:
1) y = -7x - 3.5
2) y = 1.5x + 6
3) y = -8x + 5

Quadratic functions:
1) y = -8x^2 + 4.5x - 8.5
2) y = -8.5x^2 - 10x + 3.5
3) y = -5.5x^2 - 7.5x + 3.5
4) y = -8.5x^2 + 9x - 7.5

New linear functions:
1) y = -4x - 0.5
2) y = 4.5x + 0
3) y = -5x + 8

New quadratic functions:
1) y = -10x^2 + 2.5x - 10.5
2) y = -10.5x^2 - 12x + 1.5
3) y = -7.5x^2 - 9.5x + 1.5
4) y = -10.5x^2 + 7x - 9.5

Enter x: 5

Values of the linear functions:
1) y = -20.5
2) y = 31.5
3) y = -17

Values of the quadratic functions:
1) y = -248
2) y = -321
3) y = -233.5
4) y = -237

The maximum value has function y = 4.5x + 9
Value of this function is 31.5

Press any key to continue . . .
```

Програма на Python

Header

```
1  from abc import ABC, abstractmethod
2
3  class TFunction:
4      @abstractmethod
5      def increase(number):
6          pass
7
8      @abstractmethod
9      def decrease(number):
10         pass
11
12     @abstractmethod
13     def get_value(x):
14         pass
15
16     @abstractmethod
17     def displayFunction():
18         pass
19
20 class LinearFunction(TFunction):
21     def __init__(self, A1, A0):
22         self.__A1 = A1
23         self.__A0 = A0
24
25     def increase(self, number):
26         self.__A1 += number
27         self.__A0 += number
28
29     def decrease(self, number):
30         self.__A1 -= number
31         self.__A0 -= number
32
```

```

33 ✓ def get_value(self,x):
34     y = self.__A1 * x + self.__A0
35     return y
36
37 ✓ def displayFunction(self):
38     if self.__A1!=0 and self.__A0>0:
39         print("y = ",self.__A1,"x + ",self.__A0)
40 ✓     elif self.__A1!=0 and self.__A0<0:
41         print("y = ",self.__A1,"x - ",-1*self.__A0)
42 ✓     elif self.__A1==0 and self.__A0!=0:
43         print("y = ",self.__A0)
44 ✓     elif self.__A1!=0 and self.__A0==0:
45         print("y = ",self.__A1,"x")
46 ✓     elif self.__A1==0 and self.__A0==0:
47         print("y = 0")
48
49 ✓ class QuadraticFunction(TFunction):
50 ✓     def __init__(self,B2,B1,B0):
51         self.__B2 = B2
52         self.__B1 = B1
53         self.__B0 = B0
54
55 ✓     def increase(self,number):
56         self.__B2 += number
57         self.__B1 += number
58         self.__B0 += number
59
60 ✓     def decrease(self,number):
61         self.__B2 -= number
62         self.__B1 -= number
63         self.__B0 -= number

```

```

65 ✓ def get_value(self,x):
66     y = self.__B2 * x**2 + self.__B1 * x + self.__B0
67     return y
68
69 ✓ def displayFunction(self):
70     if self.__B2!=0 and self.__B1>0 and self.__B0>0:
71         print("y = ",self.__B2,"x^2 + ",self.__B1,"x + ",self.__B0)
72 ✓     elif self.__B2!=0 and self.__B1>0 and self.__B0<0:
73         print("y = ",self.__B2,"x^2 + ",self.__B1,"x - ",-1*self.__B0)
74 ✓     elif self.__B2!=0 and self.__B1<0 and self.__B0>0:
75         print("y = ",self.__B2,"x^2 - ",-1*self.__B1,"x + ",self.__B0)
76 ✓     elif self.__B2!=0 and self.__B1<0 and self.__B0<0:
77         print("y = ",self.__B2,"x^2 - ",-1*self.__B1,"x - ",-1*self.__B0)
78 ✓     elif self.__B2==0 and self.__B1!=0 and self.__B0>0:
79         print("y = ",self.__B1,"x + ",self.__B0)
80 ✓     elif self.__B2==0 and self.__B1!=0 and self.__B0<0:
81         print("y = ",self.__B1,"x - ",-1*self.__B0)
82 ✓     elif self.__B2!=0 and self.__B1==0 and self.__B0>0:
83         print("y = ",self.__B2,"x^2 + ",self.__B0)
84 ✓     elif self.__B2!=0 and self.__B1==0 and self.__B0<0:
85         print("y = ",self.__B2,"x^2 - ",-1*self.__B0)
86 ✓     elif self.__B2!=0 and self.__B1>0 and self.__B0==0:
87         print("y = ",self.__B2,"x^2 + ",self.__B1,"x")
88 ✓     elif self.__B2!=0 and self.__B1<0 and self.__B0==0:
89         print("y = ",self.__B2,"x^2 - ",-1*self.__B1,"x")
90 ✓     elif self.__B2!=0 and self.__B1==0 and self.__B0==0:
91         print("y = ",self.__B2,"x^2")
92 ✓     elif self.__B2==0 and self.__B1!=0 and self.__B0==0:
93         print("y = ",self.__B1,"x")
94 ✓     elif self.__B2==0 and self.__B1==0 and self.__B0!=0:
95         print("y = ",self.__B0)
96 ✓     elif self.__B2==0 and self.__B1==0 and self.__B0==0:
97         print("y = 0")

```


Main

```
1  from classes import LinearFunction, QuadraticFunction
2  import random
3
4  n = int(input("Enter amount of the linear functions: "))
5  m = int(input("Enter amount of the quadratic functions: "))
6
7  linear = []
8  quadratic = []
9
10 print("\nLinear functions:")
11 for i in range(n):
12     a1 = random.randint(-20,20)
13     a0 = random.randint(-20,20)
14     linear_function = LinearFunction(a1,a0);
15     linear.append(linear_function)
16     print(i+1, ")", end = "")
17     linear_function.displayFunction()
18
19 print("\nQuadratic functions:")
20 for i in range(m):
21     b2 = random.randint(-20,20)
22     b1 = random.randint(-20,20)
23     b0 = random.randint(-20,20)
24     quadratic_function = QuadraticFunction(b2,b1,b0);
25     quadratic.append(quadratic_function)
26     print(i+1, ")", end = "")
27     quadratic_function.displayFunction()
28
29 print("\nNew linear functions:")
30 for i in range(n):
31     linear[i].increase(3)
32     print(i+1, ")", end = "")
33     linear[i].displayFunction()
34
35 print("\nNew quadratic functions:")
36 for i in range(m):
37     quadratic[i].decrease(2)
38     print(i+1, ")", end = "")
39     quadratic[i].displayFunction()
```

```

41 x = float(input("\nEnter x: "))
42
43 max_linear = linear[0].get_value(x)
44 pos_linear = 0
45
46 print("\nValues of the linear functions:")
47 for i in range(n):
48     print(i+1, ")", linear[i].get_value(x))
49     if linear[i].get_value(x) > max_linear:
50         max_linear = linear[i].get_value(x)
51         pos_linear = i
52
53 max_quadratic = quadratic[0].get_value(x)
54 pos_quadratic = 0
55
56 print("\nValues of the quadratic functions:")
57 for i in range(m):
58     print(i+1, ")", quadratic[i].get_value(x))
59     if quadratic[i].get_value(x) > max_quadratic:
60         max_quadratic = quadratic[i].get_value(x)
61         pos_quadratic = i
62
63 if max_linear > max_quadratic:
64     print("The maximum value has function ", end = "")
65     linear[pos_linear].displayFunction()
66     print("Value of this function is ", linear[pos_linear].get_value(x))
67 else:
68     print("The maximum value has function ", end = "")
69     quadratic[pos_quadratic].displayFunction()
70     print("Value of this function is ", quadratic[pos_quadratic].get_value(x))
71
72 input("\nPress Enter to continue...\n")

```

Тестування програми

```

C:\Windows\py.exe
Enter amount of the linear functions: 3
Enter amount of the quadratic functions: 4

Linear functions:
1 )y = 17 x - 2
2 )y = -7 x + 20
3 )y = 2 x + 5

Quadratic functions:
1 )y = -12 x^2 + 9 x - 19
2 )y = 9 x^2 + 10 x + 11
3 )y = -3 x^2 - 18 x + 11
4 )y = 7 x^2 + 12 x - 13

New linear functions:
1 )y = 20 x + 1
2 )y = -4 x + 23
3 )y = 5 x + 8

New quadratic functions:
1 )y = -14 x^2 + 7 x - 21
2 )y = 7 x^2 + 8 x + 9
3 )y = -5 x^2 - 20 x + 9
4 )y = 5 x^2 + 10 x - 15

Enter x: 5

Values of the linear functions:
1 ) 101.0
2 ) 3.0
3 ) 33.0

Values of the quadratic functions:
1 ) -336.0
2 ) 224.0
3 ) -216.0
4 ) 160.0

The maximum value has function y = 7 x^2 + 8 x + 9
Value of this function is 224.0
Press Enter to continue...

```