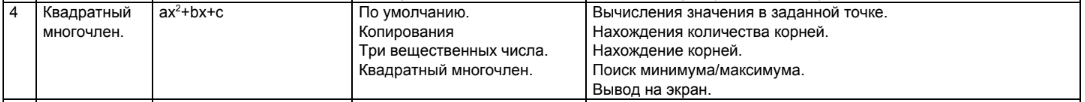
Лабараторная работа 2

Клишевич Вадим, M3105

Main.cpp

#include <iostream>  
#include "SquarePolynomial.h"  
  
using namespace std;  
  
int main() {  
 cout << "Enter polynom\n";  
 string s;  
 getline(cin, s);  
 Polynom x = Polynom(s);  
 x.print();  
  
 float solveCoordinate = 2;  
 cout << "x = " << solveCoordinate << ", y = " << x.solve(solveCoordinate) << "\n";  
  
 cout << "Roots quantity = " << x.solveCount() << "\n";  
  
 auto p = x.solve();  
 cout << "Roots: ";  
 for (auto &ans : p) {  
 cout << ans << ", ";  
 }  
 cout << "\n";  
  
 cout << "min = " << x.findMin() << "\n" << "max = " << x.findMax() << "\n";  
  
}

SquarePolynomial.cpp

//  
// Created by vadim on 5.02.20.  
//  
  
#include "SquarePolynomial.h"  
  
Polynom::Polynom() {  
 srand(time(nullptr));  
 a = static\_cast<float>(rand()) / static\_cast<float>(RAND\_MAX) \* 10;  
 b = static\_cast<float>(rand()) / static\_cast<float>(RAND\_MAX) \* 10;  
 c = static\_cast<float>(rand()) / static\_cast<float>(RAND\_MAX) \* 10;  
}  
  
Polynom::Polynom(const float &a\_, const float &b\_, const float &c\_) {  
 a = a\_;  
 b = b\_;  
 c = c\_;  
}  
  
Polynom::Polynom(const string &s) {  
 regex re("(([+-]?)\\s?([0-9]\*\\.[0-9]+|[0-9]+)\\s?\\\*?\\s?x\\^\\s?2\\s?)?"  
 "(([+-]?)\\s?([0-9]\*\\.[0-9]+|[0-9]+)\\s?\\\*?\\s?x\\s?)?"  
 "(([+-]?)\\s?([0-9]\*\\.[0-9]+|[0-9]+))?");  
 smatch match;  
 if (regex\_search(s, match, re) && match.size() > 6) {  
 if (!match.str(1).empty()) {  
 a = stof(match.str(3));  
 if (match.str(2) == "-") {  
 a \*= -1;  
 }  
 } else {  
 a = 0;  
 }  
 if (!match.str(4).empty()) {  
 b = stof(match.str(6));  
 if (match.str(5) == "-") {  
 b \*= -1;  
 }  
 } else {  
 b = 0;  
 }  
 if (!match.str(7).empty()) {  
 c = stof(match.str(9));  
 if (match.str(8) == "-") {  
 c \*= -1;  
 }  
 } else {  
 c = 0;  
 }  
 } else {  
 cout << "Cannot compile\n";  
 exit(123);  
 }  
}  
  
Polynom::Polynom(const Polynom &x) {  
 a = x.a;  
 b = x.b;  
 c = x.c;  
}  
  
void Polynom::print() {  
 cout << "Polynom: (" << a << ") \* x^2 + (" << b << ") \* x + (" << c << ")\n\n";  
}  
  
float Polynom::solve(const float &x) {  
 return a \* x \* x + b \* x + c;  
}  
  
vector<float> Polynom::solve() {  
 if (fabs(a) > 0.000001) {  
 if (b \* b - a \* c < 0) {  
 return {};  
 }  
 float D = sqrt(b \* b - a \* c);  
 float x1 = (-b - D) / 2 / a;  
 float x2 = (-b + D) / 2 / a;  
 if (fabs(x1 - x2) < 0.000001) {  
 return {fabs(x1)};  
 } else {  
 return {x1, x2};  
 }  
 } else if (fabs(b) > 0.000001) {  
 return {-c / b};  
 } else {  
 return {};  
 }  
}  
  
int Polynom::solveCount() {  
 if (fabs(a) > 0.000001) {  
 float D = b \* b - a \* c;  
 if (D < -0.000001) return 0;  
 else if (D > 0.000001) return 2;  
 else return 1;  
 } else if (fabs(b) > 0.000001) {  
 return 1;  
 } else {  
 if (fabs(c) > 0.000001) {  
 return 0;  
 } else {  
 return INT\_MAX;  
 }  
 }  
}  
  
float Polynom::findMax() {  
 if (fabs(a) > 0.000001) {  
 Polynom der = Polynom(0, a, b);  
 float temp = der.solve()[0];  
 if (a > 0) {  
 return MAXFLOAT;  
 } else {  
 return solve(temp);  
 }  
 } else if (fabs(b) > 0.000001) {  
 return MAXFLOAT;  
 } else {  
 return c;  
 }  
}  
  
float Polynom::findMin() {  
 if (fabs(a) > 0.000001) {  
 Polynom der = Polynom(0, a, b);  
 float temp = der.solve()[0];  
 if (a > 0) {  
 return solve(temp);  
 } else {  
 return -MAXFLOAT;  
 }  
 } else if (fabs(b) > 0.000001) {  
 return -MAXFLOAT;  
 } else {  
 return c;  
 }  
}

SquarePolynomial.h

//  
// Created by vadim on 5.02.20.  
//  
  
#include <bits/stdc++.h>  
#include <string>  
  
using namespace std;  
  
#pragma once  
  
class Polynom {  
private:  
 float a, b, c;  
  
public:  
 Polynom();  
  
 Polynom(const float &, const float &, const float &);  
  
 Polynom(const string &);  
  
 Polynom(const Polynom &);  
  
 void print();  
  
 float solve(const float &);  
  
 vector<float> solve();  
  
 int solveCount();  
  
 float findMin();  
  
 float findMax();  
};