

Міністерство освіти і науки України

КПІ ім. Ігоря Сікорського

Кафедра ІІІ

ЗВІТ

з виконання лабораторної роботи № 7

з кредитного модуля

“Основи програмування-2. Методології програмування”

Варіант № 22

Виконав:

студент 1-го курсу

гр. ІІ-22 ФІОТ

Підпанюк Віталій Андрійович

Київ 2023

Завдання:

22. Розробити клас «Функція», що представляє собою функцію виду $y(x) = \left| \frac{|x-2|+1}{x-2} \right|$. Членом-даним класу є аргумент x , а одним із методів - обчислення значення функції для заданого аргумента. Обчислити значення функції на заданому проміжку із заданим кроком. Для значень, які не задовільняють області визначення функції, згенерувати відповідні виняткові ситуації і організувати їх обробку.

Код програми:

main.cpp:

```
#include "mainwindow.h"
```

```
#include <QApplication>
```

```
int main(int argc, char *argv[]) {
```

```
    QApplication app(argc, argv);
```

```
    NumberRangeWidget widget;
```

```
    widget.show();
```

```
    return app.exec();
```

```
}
```

mainwindow.cpp:

```
#include "mainwindow.h"
```

```
#include "inequality.h"
```

```
#include <QLabel>
```

```
#include <QLineEdit>
```

```
#include <QPushButton>
```

```
#include <QTextEdit>
```

```
#include <QVBoxLayout>
```

```
NumberRangeWidget::NumberRangeWidget(QWidget* parent) :  
QWidget(parent) {
```

```
    lowerLabel = new QLabel("Нижня межа:");  
    upperLabel = new QLabel("Верхня межа:");  
    stepLabel = new QLabel("Крок");  
    lowerLineEdit = new QLineEdit;  
    upperLineEdit = new QLineEdit;  
    stepLineEdit = new QLineEdit;  
    processButton = new QPushButton("Обробити");  
    outputTextEdit = new QTextEdit;
```

```
    QVBoxLayout* layout = new QVBoxLayout;  
    layout->addWidget(lowerLabel);  
    layout->addWidget(lowerLineEdit);  
    layout->addWidget(upperLabel);  
    layout->addWidget(upperLineEdit);  
    layout->addWidget(stepLabel);  
    layout->addWidget(stepLineEdit);  
    layout->addWidget(processButton);  
    layout->addWidget(outputTextEdit);
```

```
    setLayout(layout);
```

```
    connect(processButton, &QPushButton::clicked, this,  
    &NumberRangeWidget::handleProcessButtonClick);  
}
```

```

void NumberRangeWidget::handleProcessButtonClick() {
    // Отримати значення нижньої та верхньої межі
    int lower = lowerLineEdit->text().toInt();
    int upper = upperLineEdit->text().toInt();
    int step = stepLineEdit->text().toInt();

    QString result;
    for (double num = lower; num <= upper; num += step) {
        QString error_message;
        try
        {
            equality unit(num);
            result += QString::number(unit.solve(num), 'f', 4) + '\n';
        }
        catch(Error1& zero)
        {
            error_message = zero.message();
            result += QString(error_message);
        }
        catch(Error2& infinity)
        {
            error_message = infinity.message();
            result += QString(error_message);
        }

    }
    outputTextEdit->setText(result);
}

```

mainwindow.h:

#ifndef NUMBERRANGEWIDGET_H

#define NUMBERRANGEWIDGET_H

#include <QWidget>

class QLabel;

class QLineEdit;

class QPushButton;

class QTextEdit;

class NumberRangeWidget : public QWidget {

Q_OBJECT

public:

explicit NumberRangeWidget(QWidget* parent = nullptr);

private:

QLabel* lowerLabel;

QLabel* upperLabel;

QLabel* stepLabel;

QLineEdit* lowerLineEdit;

QLineEdit* upperLineEdit;

QLineEdit* stepLineEdit;

QPushButton* processButton;

QTextEdit* outputTextEdit;

```
private slots:
    void handleProcessButtonClick();
};

#endif // NUMBERRANGEWIDGET_H
```

```
inequeliy.h:
#ifndef INEQUALITY_H
#define INEQUALITY_H

#include<iostream>
#include <cmath>
#include "mainwindow.h"
using namespace std;

class Error1
{
    int m;
    QString error_text = "Devide by zero\n";
public:
    Error1(int x) : m(x){}
    virtual QString message()
    {
        return error_text;
    }
};
```

```

class Error2
{
    int m;
    QString error_text = "Devide by infinity\n";
public:
    Error2(int x) : m(x){}
    virtual QString message()
    {
        return error_text;
    }

};

```

```

class equality
{
    double x;
public:
    equality(int a)
    {
        if((a - 2)==0)
        {
            throw Error1(a);
        }
        else
        {
            if(abs(a)>=2000000)
            {

```

```
        throw Error2(a);
    }
    else
    {
        x = a;
    }
}
}
double solve(double a)
{
    x = a;
    double y = abs((abs(x-2)+1)/(x-2));
    return y;
}
};
```

```
void main_program();
#endif // INEQUALITY_H
```