Міністерство освіти і науки України

Запорізький національний технічний університет

Кафедра програмних засобів

ЗВІТ

З лабораторної роботи №3

з дисципліни «Якість програмного забезпечення та тестування» на тему:

«ОБЧИСЛЕННЯ МЕТРИК РОЗМІРУ ТА СКЛАДНОСТІ ПРОГРАМ»

Виконав:

ст. гр. КНТ-415 К. А. Калоян

Прийняв:

к.т.н.,доцент Т. І. Каплієнко

2018

**Виконання лабораторної роботи:**

Код програми

#include "mainwindow.h"

#include "ui\_mainwindow.h"

#include <QString>

#include <QLocale>

#include <QStringList>

#include <QList>

#include <QRegExp>

#include <qmath.h>

MainWindow::MainWindow(QWidget \*parent) :

QMainWindow(parent),

ui(new Ui::MainWindow)

{

ui->setupUi(this);

}

MainWindow::~MainWindow()

{

delete ui;

}

int n1, n2, m, N1, N2, N, R;

// Рассчитываем общее количество строк

void MainWindow::on\_SLOC\_clicked()

{

QString str= ui->plainTextEdit->toPlainText();

int n = str.count("\n");

ui->label->setText(QString::number(n+1));

}

// Рассчитываем количество пустых строк

void MainWindow::on\_EmptyLines\_clicked()

{

QString str= ui->plainTextEdit->toPlainText();

int n = str.count("\n\n");

ui->label\_2->setText(QString::number(n));

}

// Рассчитываем количество строк, содержащих комментарии

void MainWindow::on\_Comments\_clicked()

{

QString str= ui->plainTextEdit->toPlainText();

int n = str.count("\n//");

ui->label\_3->setText(QString::number(n));

int k = str.count("\n");

ui->label\_5->setText(QString::number(((float) n)/(k+1)\*100)+"%");

}

// Рассчитываем количество строк кода (общее количество строк - комментарии - пустые строки)

void MainWindow::on\_OnlyCode\_clicked()

{

QString str = ui->plainTextEdit->toPlainText();

int n = str.count("\n");

int k = str.count("\n//");

ui->label\_4->setText(QString::number(n-k-str.count("\n\n")));

}

// Рассчитываем словарь операторов (количество уникальных операторов программы, также символы-разделители, имена процедур и знаки операций)

void MainWindow::on\_VocOperatn1\_clicked()

{

QString str= ui->plainTextEdit->toPlainText();

QStringList list = str.split(" ");

QString types = "+ - = += ++ -- \* << >> < > != == || && &";

QStringList lst = str.split(QRegExp("(void|int|string|float|double|byte)[A-ZА-z0-9]\*"), QString::SkipEmptyParts);

QStringList ltypes;

QStringList ress;

list.removeDuplicates ();

ltypes = types.split(" ");

QStringList::Iterator itt = ltypes.begin();

for(int i = list.count()-1; i >= 0; --i)

{

const QString& item = list[i];

itt = ltypes.begin();

while (itt != ltypes.end()){

if(item==\*itt){

// list.removeAt(i);

ress << item;

}

++itt;

}

}

//===========================

int r=0;

QStringList::iterator t = lst.begin();

for(int i = lst.count()-1; i >= 0; --i)

{

const QString& item = lst[i];

r = r+item.count(")\n");

}

R = r;

n1 = ress.count();

ui->label\_7->setText(QString::number(ress.count()+r));

}

// Рассчитываем словарь операндов (количество уникальных операндов программы)

void MainWindow::on\_VocOperandn2\_clicked()

{

QString str= ui->plainTextEdit->toPlainText();

QRegExp rx("([\\w\\']+)[\\s,.;]");

QStringList list;

int pos = 0;

while ((pos = rx.indexIn(str, pos)) != -1) {

list << rx.cap(1);

pos += rx.matchedLength();

}

list.removeDuplicates ();

QString types = "int void string double float iterator List const array for while do goto";

QStringList ltypes;

ltypes = types.split(" ");

QStringList::Iterator itt = ltypes.begin();

for(int i = list.count()-1; i >= 0; --i)

{

const QString& item = list[i];

itt = ltypes.begin();

while (itt != ltypes.end()){

if(item==\*itt)

list.removeAt(i);

++itt;}

if ((item.toFloat()!=0)) {list.removeAt(i);}

}

for(int i = list.count()-1; i >= 0; --i)

{

const QString& item = list[i];

if (item=="0") {list.removeAt(i);}

}

n2=list.count();

ui->label\_6->setText(QString::number(list.count()));

}

// Рассчитываем общее количество операторов в программе

void MainWindow::on\_AllOperatN1\_clicked()

{

QString str= ui->plainTextEdit->toPlainText();

QStringList list = str.split(" ");

QString types = "+ - = += ++ -- \* << >> < > != == || && &";

QStringList ltypes;

QStringList ress;

ltypes = types.split(" ");

QStringList::Iterator itt = ltypes.begin();

for(int i = list.count()-1; i >= 0; --i)

{

const QString& item = list[i];

itt = ltypes.begin();

while (itt != ltypes.end()){

if(item==\*itt){

ress << item;

}

++itt;

}

}

N1 = ress.count();

ui->label\_8->setText(QString::number(ress.count()+R));

}

// Рассчитываем общее количество операндов в программе

void MainWindow::on\_AllOperandN2\_clicked()

{

QString str= ui->plainTextEdit->toPlainText();

QRegExp rx("([\\w\\']+)[\\s,.;]");

QStringList list;

int pos = 0;

while ((pos = rx.indexIn(str, pos)) != -1) {

list << rx.cap(1);

pos += rx.matchedLength();

}

QString types = "int void string double float iterator List const array for while do goto";

QStringList ltypes;

ltypes = types.split(" ");

QStringList::Iterator itt = ltypes.begin();

for(int i = list.count()-1; i >= 0; --i)

{

const QString& item = list[i];

itt = ltypes.begin();

while (itt != ltypes.end()){

if(item==\*itt)

list.removeAt(i);

++itt;}

if ((item.toFloat()!=0)) {list.removeAt(i);}

}

for(int i = list.count()-1; i >= 0; --i)

{

const QString& item = list[i];

if (item=="0") {list.removeAt(i);}

}

N2 = list.count();

ui->label\_9->setText(QString::number(list.count()));

}

// Рассчитываем длину программы N = N1 + N2 + R

void MainWindow::on\_N\_clicked()

{

N = N1+N2+R;

ui->label\_12->setText(QString::number(N));

}

// Рассчитываем объем программы N\*log2m

void MainWindow::on\_V\_clicked()

{

float q = N\*qLn(m)/qLn(2);

ui->label\_13->setText(QString::number(q));

}

// Рассчитываем цикломатическую сложность

void MainWindow::on\_Cyclomatic\_clicked()

{

QString str= ui->plainTextEdit->toPlainText();

QStringList list = str.split(QRegExp("(for|while)"), QString::SkipEmptyParts);

QStringList::Iterator it = list.begin();

int s=0;

for(int i = list.count()-1; i >= 0; --i)

{

const QString& item = list[i];

if ((item.count("{")>=item.count("}"))&&(item.count("}")!=0)) {s=s+1; /\*ui->plainTextEdit->insertPlainText(\*it+"+++");\*/}

}

ui->label\_10->setText(QString::number(s+str.count("if(")));

}

// словарь программы m = n1 + n2 + R

void MainWindow::on\_Voc\_clicked()

{

m = n1+ n2+R;

ui->label\_11->setText(QString::number(m));

}

// Абсолютная сложность по Джилбу - количество условных операторов

void MainWindow::on\_CI\_clicked()

{

QString str= ui->plainTextEdit->toPlainText();

int n = str.count("if(");

ui->label\_14->setText(QString::number(n));

}

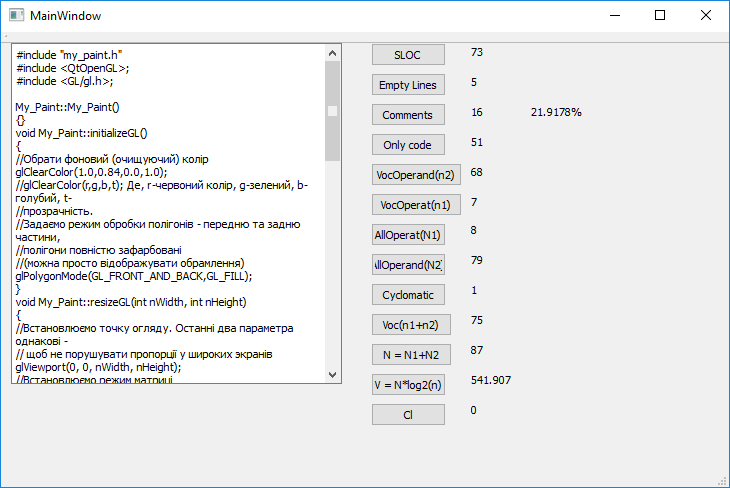


Рисунок 1 – Виконання програми