Міністерство освіти і науки України

Запорізький національний технічний університет

кафедра програмних засобів

ЗВІТ

З лабораторної роботи №3

З дисципліни «Якість програмного забезпечення та тестування» з теми:

**«ОБЧИСЛЕННЯ МЕТРИК РОЗМІРУ ТА СКЛАДНОСТІ ПРОГРАМ»**

Виконав:

студент групи КНТ-115 Ю.О. Лукашенко

Прийняла:

доцент, к.т.н Н.О.Миронова

2018

1. Мета роботи

Вивчити основні метрики коду програмних систем та реалізувати їх із застосуванням мови С++ та Qt.

1. Завдання до роботи

Реалізувати на мові C++ з використанням крос-платформного інструментарію Qt програму розрахунку метрик:

* кількість рядків коду;
* кількість пустих рядків;
* кількість рядків, що містять коментарі;
* кількість рядків, що містять вихідний код і коментарі;
* відсоток коментарів;
* словник операторів (кількість унікальних операторів програми, також символи-роздільники, імена процедур і знаки операцій);
* словник операндів (кількість унікальних операндів програми);
* загальна кількість операторів в програмі;
* загальна кількість операндів в програмі;
* довжина програми;
* обсяг програми;
* цикломатична складність;
* максимальний рівень вкладеності.

1. Хід роботи

***3.3.1 Файл mainwindow.h***

#ifndef MAINWINDOW\_H

#define MAINWINDOW\_H

#include <QMainWindow>

namespace Ui {

class MainWindow;

}

class MainWindow : public QMainWindow

{

Q\_OBJECT

public:

explicit MainWindow(QWidget \*parent = 0);

~*MainWindow*();

private slots:

void on\_SLOC\_clicked();

void on\_BLOC\_clicked();

void on\_CLOC\_clicked();

void on\_CSLOC\_clicked();

void on\_PerComment\_clicked();

void on\_VocOperands\_clicked();

void on\_AllOperands\_clicked();

void on\_VocOperats\_clicked();

void on\_ALLOperats\_clicked();

void on\_Cyclomatic\_clicked();

void on\_N\_clicked();

void on\_V\_clicked();

void on\_Voc\_clicked();

void on\_Nesting\_clicked();

private:

Ui::MainWindow \*ui;

};

#endif // MAINWINDOW\_H

***3.3.2 Файл main.cpp***

#include "mainwindow.h"

#include <QApplication>

int main(int argc, char \*argv[])

{

QApplication a(argc, argv);

MainWindow w;

w.show();

return a.exec();

}

***3.3.3 Файл mainwindow.cpp***

#include "mainwindow.h"

#include "ui\_mainwindow.h"

#include "qmath.h"

int d, t, D, T, voc, N;

MainWindow :: MainWindow(QWidget \* parent) :

QMainWindow(parent),

ui(new Ui :: MainWindow)

{

ui -> setupUi(this);

}

MainWindow :: ~ *MainWindow*()

{

delete ui;

}

//кол-во строк кода (source lines of code)

void MainWindow :: on\_SLOC\_clicked()

{

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

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

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

}

//кол-во пустых строк (blank lines of code)

void MainWindow::on\_BLOC\_clicked()

{

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

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

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

}

//кол-во комментариев (comment lines of code)

void MainWindow::on\_CLOC\_clicked()

{

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

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

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

}

//кол-во комментариев и кода (Lines with Both Code and Comments)

void MainWindow::on\_CSLOC\_clicked()

{

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

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

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

}

//процент коментариев

void MainWindow::on\_PerComment\_clicked()

{

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

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

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

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

}

//словарь операндов

void MainWindow::on\_VocOperands\_clicked()

{

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

QString res;

QStringList cur = str.split("\n");

for(int i=0; i<cur.size(); i++)

{

if(cur[i].startsWith("//") == false)

{

res += cur[i];

}

}

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

QStringList list;

int pos = 0;

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

{

list << rx.cap(1);

pos += rx.matchedLength();

}

list.removeDuplicates();

QString types = "int void string double float new delete Iterator List const array if else for while MainWindow QWidget QMainWindow Ui QRegExp QStringList QString";

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);

}

}

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

ui->vocoperandsBox->setPlainText(list.join("\n"));

d = list.count();

}

//словарь операторов

void MainWindow::on\_VocOperats\_clicked()

{

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

QStringList cur = str.split("\n");

QString res;

for(int i=0; i<cur.size(); i++)

{

if(cur[i].startsWith("//") == false && cur[i].startsWith("\"") == false)

{

res += cur[i];

}

}

QStringList list = res.split(" ");

QString types =

"+ - = += ++ -- \* << >> < > != == || && & % <= >= ! ~ | ^ -= \*= /= . -> , ? :: new delete int void string double float new delete Iterator List const array if else for while MainWindow QWidget QMainWindow Ui QRegExp QStringList QString";

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)

{

ress << item;

}

++ itt;

}

}

ress.removeDuplicates();

ui->label\_vocoperats->setText(QString::number(ress.count()));

ui->vocoperatsBox->setPlainText(ress.join("\n"));

t = ress.count();

}

//общее кол-во операторов

void MainWindow::on\_ALLOperats\_clicked()

{

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

QStringList list = str.split(" ");

QString types =

"+ - = += ++ -- \* << >> < > != == || && & % <= >= ! ~ | ^ -= \*= /= . -> , ? :: new delete int void string double float new delete Iterator List const array if else for while MainWindow QWidget QMainWindow Ui QRegExp QStringList QString";

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;

}

}

ui->label\_alloperats->setText(QString::number(ress.count()));

T = ress.count();

}

//общее кол-во операндов

void MainWindow::on\_AllOperands\_clicked()

{

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

QString res;

QStringList cur = str.split("\n");

for(int i=0; i<cur.size(); i++)

{

if(cur[i].startsWith("//") == false)

{

res += cur[i];

}

}

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

QStringList list;

int pos = 0;

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

{

list << rx.cap(1);

pos += rx.matchedLength();

}

QString types = "int void string double float new delete Iterator List const array if else for while MainWindow QWidget QMainWindow Ui QRegExp QStringList QString";

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);

}

}

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

D = list.count();

}

//Цикломатическая сложность

void MainWindow::on\_Cyclomatic\_clicked()

{

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

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

int s = list.count()-1;

ui->label\_cyclomatic->setText(QString::number(s));

}

//словарь программы

void MainWindow::on\_Voc\_clicked()

{

voc = t + d;

ui->label\_hpvoc->setText(QString::number(voc));

}

//Длина программы

void MainWindow::on\_N\_clicked()

{

N = T + D;

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

}

//Объём программы N\*log2voc

void MainWindow::on\_V\_clicked()

{

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

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

}

//Максимальный уровень вложенности

void MainWindow::on\_Nesting\_clicked()

{

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

int max = 0;

int cur = 0;

QStringList res;

res = str.split(" ");

for(int i = 0; i<res.size()-1; i++)

{

if(res[i].startsWith("if(") || res[i].startsWith("for(") || res[i].startsWith("while(") )

{

++cur;

if(max < cur)

{

if(true)

{

max = cur; //test for max nesting level

}

}

}

else if (res[i].startsWith("}"))

{

if(cur>0)

{

cur--;

}

}

}

ui->label\_nesting->setText(QString::number(max));

}

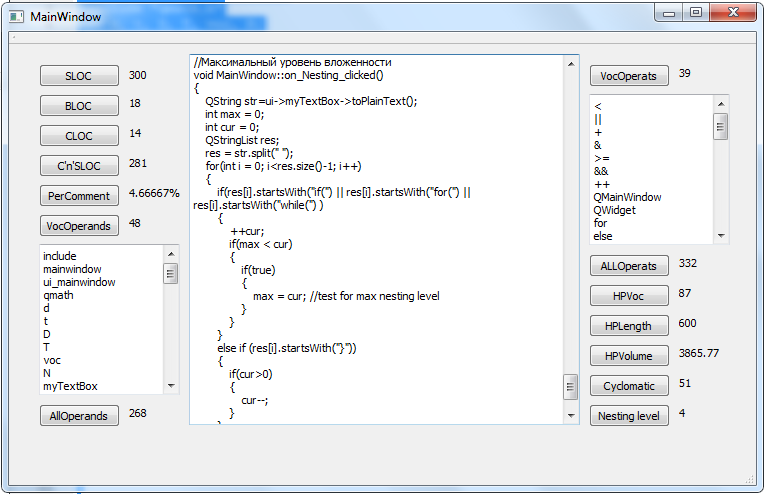


Рисунок 3.1 – Виконання програми

1. Висновки

В ході виконання лабораторної роботи, я засвоїв основні метрики коду програмних систем та реалізував їх із застосуванням мови С++ та Qt.