**Міністерство освіти і науки України**

**Запорізький національний технічний університет**

Кафедра програмних засобів

**ЗВІТ**

З лабораторної роботи №3

з дисципліни «ЯПЗтаТ»

на тему:

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

|  |  |
| --- | --- |
| Виконав: |  |
| студент КНТ-415 | Р.О. Шлома |
|  |  |
| Прийняв: |  |
| доцент | Т. І. Каплієнко |
|  |  |
| 2018 | |

Мета

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

Завдання до роботи

1. Ознайомитися з теоретичними відомостями щодо метрик програмних систем.
2. Реализувати 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 і довжина програми.
3. Виконати аналіз отриманих результатів.

Виконання завдань до роботи

*Код розробленої програми*

#include "mainwindow.h"

#include "ui\_mainwindow.h"

#include <QString>

#include <QLocale>

#include <QStringList>

#include <QList>

#include <QRegExp>

#include <qmath.h>

#include <QFileDialog>

#include <iostream>

#include <fstream>

#include <QMessageBox>

using namespace std;

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

MainWindow::**MainWindow**(QWidget \*parent) :

QMainWindow(parent),

ui(new Ui::MainWindow)

{

ui->setupUi(this);

connect(ui->loadAction, SIGNAL(triggered(bool)), this, SLOT(loadFile()));

connect(ui->buttonAnalyze, SIGNAL(clicked()), this, SLOT(analyzationAll()));

}

MainWindow::~***MainWindow***()

{

delete ui;

}

// количество строк кода

void MainWindow::**on\_SLOC\_clicked**()

{

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

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

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

}

//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));

}

// количество коментареев и в процентах

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) + "%");

}

//общее количество строк кода

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")));

}

//словарь операторов (кількість унікальних операторів програми,

//також символи-роздільники, імена процедур і знаки операцій)

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-ZРђ-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));

}

//словарь операндов(кількість унікальних операндів програми)

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()));

}

// общее количество операторов

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));

}

// общее количество операндов

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()));

}

//длина программы

void MainWindow::**on\_N\_clicked**()

{

N = N1+N2+R;

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

}

//объем программы

void MainWindow::**on\_V\_clicked**()

{

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

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

}

// цикломатическая 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(")));

}

//максимальный уровень вложености программы

void MainWindow::**on\_Voc\_clicked**()

{

m = n1+ n2+R;

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

}

//длина программы

void MainWindow::**on\_CI\_clicked**()

{

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

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

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

}

//загрузка файла с помощью диалогового окна

void MainWindow::**loadFile**()

{

QString fileName = QFileDialog::getOpenFileName(0, "Open Dialog File", "", "\*.cpp \*.h");

ifstream file(fileName.toStdString());

char \*buffer = new char [50];

if(file.is\_open() == false)

{

QMessageBox \*mesBox = new QMessageBox();

mesBox->setText("File could not open!");

mesBox->show();

}

else

{

while(!file.eof())

{

for(int i = 0; i < 50; i++)

buffer[i] = NULL;

file.read(buffer, 50);

cout << buffer;

ui->plainTextEdit->setPlainText(ui->plainTextEdit->toPlainText() + QString::fromLatin1(buffer));

}

}

file.close();

}

void MainWindow::**analyzationAll**()

{

on\_SLOC\_clicked();

on\_EmptyLines\_clicked();

on\_Comments\_clicked();

on\_OnlyCode\_clicked();

on\_VocOperandn2\_clicked();

on\_VocOperatn1\_clicked();

on\_AllOperatN1\_clicked();

on\_AllOperandN2\_clicked();

on\_Cyclomatic\_clicked();

on\_Voc\_clicked();

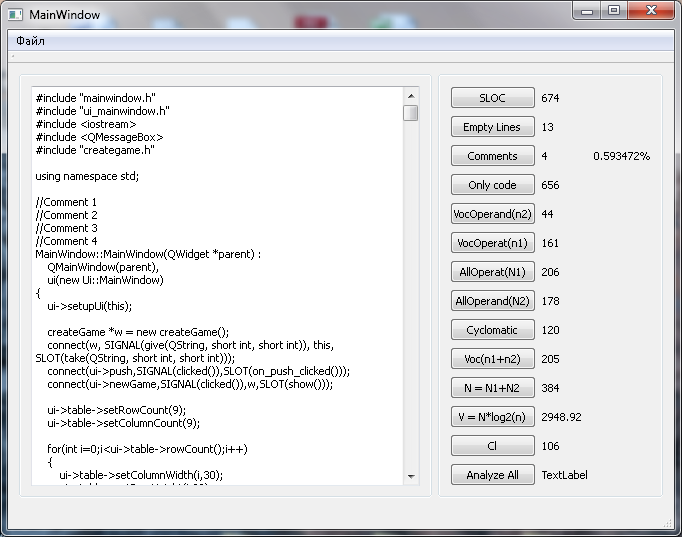
on\_N\_clicked();

on\_V\_clicked();

on\_CI\_clicked();

}

*Результат роботи програми*



Висновок

Вивчив основні метрики коду програмних систем.