**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра Математического обеспечения и применения ЭВМ**

отчет

**по лабораторной работе № 1**

**по дисциплине «Алгоритмы и Структуры Данных»**

Тема: **Рекурсия**

|  |  |  |
| --- | --- | --- |
| Студент гр. 6304 |  | Ковынев М.В. |
| Преподаватель |  | Балтрашевич В.Э. |

Санкт-Петербург

2017

# **Цель работы**

Ознакомиться с основными понятиями и приёмами рекурсивного программирования, получить навыки программирования рекурсивных процедур и функций.

# **Постановка задачи**

Напечатать все перестановки заданных n различных натуральных чисел (или символов).

# **Ход работы**

1. Реализуем считывание данных (чисел) с файла, указанного пользователем при помощи следующей функции:

void MainWindow::**on\_action\_triggered**()

{

QString fileName = QFileDialog::getOpenFileName(this, tr("Open File"), QString(),

tr("Text Files (\*.txt)"));

if (!fileName.isEmpty())

{

QFile file(fileName);

if (!file.*open*(QIODevice::ReadOnly))

{

QMessageBox::critical(this, tr("Error"), tr("Could not open file"));

return;

}

QTextStream in(&file);

ui->textEdit->setText(in.readAll());

file.*close*();

}

}

1. Реализуем сохранение полученных перестановок в файл, указанный пользователем при помощи следующей функции:

void MainWindow::**on\_action\_2\_triggered**()

{

QString fileName = QFileDialog::getSaveFileName(this, tr("Save File"), QString(),

tr("Text Files (\*.txt)"));

if (!fileName.isEmpty())

{

QFile file(fileName);

if (!file.*open*(QIODevice::WriteOnly))

{

QMessageBox::critical(this, tr("Error"), tr("Could not save file"));

return;

} else

{

QTextStream stream(&file);

stream << ui->textEdit\_2->toPlainText();

stream.flush();

file.*close*();

}

}

}

1. Используем функцию *generate* для генерации всех вариантов перестановок чисел.

void MainWindow::**generatePermutations**(int k, int N, QString\* array)

{

if (k == N-1)

{

ui->textEdit\_2->setText(ui->textEdit\_2->toPlainText()+QString::number(++count)+". ");

ui->progressBar->setValue(count\*100.0/maxfact);

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

{

ui->textEdit\_2->setText(ui->textEdit\_2->toPlainText()+array[i] + " ");

}

ui->textEdit\_2->setText(ui->textEdit\_2->toPlainText()+"\n");

}

else

{

for (int j = k; j < N; j++)

{

std::swap(array[k], array[j]);

generatePermutations(k+1, N, array);

std::swap(array[k], array[j]);

}

}

}

# **Пример работы программы**

|  |  |
| --- | --- |
| **Входные данные** | **Выходные данные** |
| 1 2 3 | 1. 1 2 3  2. 1 3 2  3. 2 1 3  4. 2 3 1  5. 3 2 1  6. 3 1 2 |
| 123 222 99 0 | 1. 123 222 99 0  2. 123 222 0 99  3. 123 99 222 0  4. 123 99 0 222  5. 123 0 99 222  6. 123 0 222 99  7. 222 123 99 0  8. 222 123 0 99  9. 222 99 123 0  10. 222 99 0 123  11. 222 0 99 123  12. 222 0 123 99  13. 99 222 123 0  14. 99 222 0 123  15. 99 123 222 0  16. 99 123 0 222  17. 99 0 123 222  18. 99 0 222 123  19. 0 222 99 123  20. 0 222 123 99  21. 0 99 222 123  22. 0 99 123 222  23. 0 123 99 222  24. 0 123 222 99 |
| 555 | 1.555 |
| Qt is awesome! | 1. Qt is awesome!  2. Qt awesome! is  3. is Qt awesome!  4. is awesome! Qt  5. awesome! is Qt  6. awesome! Qt is |
| 2+2 is four | 1. 2+2 is four  2. 2+2 four is  3. is 2+2 four  4. is four 2+2  5. four is 2+2  6. four 2+2 is |

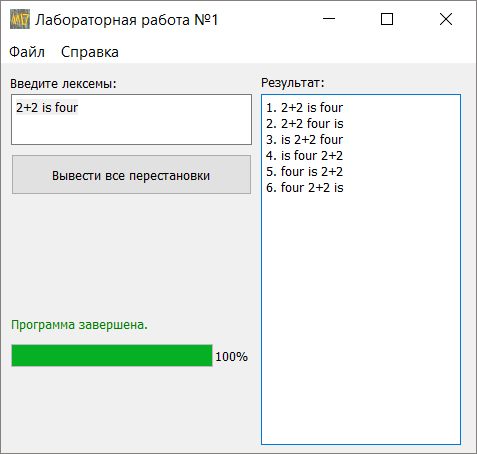


Рисунок 1. Вид программы

# **Вывод**

В ходе выполнения данное лабораторной работы были изучены основные понятия и приёмы рекурсивного программирования, были получены навыки программирования рекурсивных процедур и функций.

# **ПРИЛОЖЕНИЕ А.**

# **ИХОДНЫЙ КОД MAIN.CPP**

#include "mainwindow.h"

#include <QApplication>

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

{

QApplication a(argc, argv);

MainWindow w;

w.setWindowIcon(QIcon("C:/Downloads/2.png"));

w.show();

w.setFixedSize(475,415);

w.setWindowTitle("Лабораторная работа №1");

return a.exec();

}

# **ПРИЛОЖЕНИЕ Б.**

# **ИХОДНЫЙ КОД MAINWINDOW.CPP**

#include "mainwindow.h"

#include "ui\_mainwindow.h"

#include <QFileDialog>

#include <QFile>

#include <QMessageBox>

#include <QTextStream>

#include <QString>

#include <algorithm>

#include <QDebug>

#include "time.h"

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

QMainWindow(parent),

ui(new Ui::MainWindow)

{

ui->setupUi(this);

ui->progressBar->setValue(0);

ui->progressBar->setHidden(1);

}

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

{

delete ui;

}

void MainWindow::**on\_action\_triggered**()

{

QString fileName = QFileDialog::getOpenFileName(this, tr("Открыть файл"), QString(),

tr("Text Files (\*.txt)"));

if (!fileName.isEmpty())

{

QFile file(fileName);

if (!file.*open*(QIODevice::ReadOnly))

{

QMessageBox::critical(this, tr("Ошибка"), tr("Не могу открыть данный файл"));

return;

}

QTextStream in(&file);

ui->textEdit->setText(in.readAll());

file.*close*();

}

}

void MainWindow::**on\_action\_2\_triggered**()

{

QString fileName = QFileDialog::getSaveFileName(this, tr("Сохранить файл"), QString(),

tr("Text Files (\*.txt)"));

if (!fileName.isEmpty())

{

QFile file(fileName);

if (!file.*open*(QIODevice::WriteOnly))

{

QMessageBox::critical(this, tr("Ошибка"), tr("Не могу открыть данный файл"));

return;

} else

{

QTextStream stream(&file);

stream << ui->textEdit\_2->toPlainText();

stream.flush();

file.*close*();

}

}

}

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

{

exit(1);

}

double **fact**(int N)

{

if(N < 0)

return 0;

if (N == 0)

return 1;

else

return N \* fact(N - 1);

}

int count=0;

int maxfact;

void MainWindow::**generatePermutations**(int k, int N, QString\* array)

{

if (k == N-1)

{

ui->textEdit\_2->setText(ui->textEdit\_2->toPlainText()+QString::number(++count)+". ");

ui->progressBar->setValue(count\*100.0/maxfact);

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

{

ui->textEdit\_2->setText(ui->textEdit\_2->toPlainText()+array[i] + " ");

}

ui->textEdit\_2->setText(ui->textEdit\_2->toPlainText()+"\n");

}

else

{

for (int j = k; j < N; j++)

{

std::swap(array[k], array[j]);

generatePermutations(k+1, N, array);

std::swap(array[k], array[j]);

}

}

}

void MainWindow::**on\_pushButton\_2\_clicked**()

{

ui->textEdit\_2->clear();

ui->progressBar->setHidden(0);

QString inputString = ui->textEdit->toPlainText().simplified();

QString\* array = new QString[12];

int y=0;

int size=0;

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

{

if (inputString[i] == ' ')

{

QString x="";

for (int j=y; j < i; j++)

{

x +=inputString[j];

}

array[size]=x;

size++;

y=i+1;

}

}

QString x="";

for (int j=y; j < inputString.size(); j++)

{

x +=inputString[j];

}

array[size]=x;

size++;

maxfact = fact(size);

if (size>=6)

{

QMessageBox::critical(this, tr("Ошибка"), tr("Возможное переполнение памяти. Введите меньшее количество чисел"));

return;

}

ui->label\_3->setText("Пожалуйста, подождите.\nВыполняется генерация перестановок.");

ui->label\_3->setStyleSheet("QLabel {color : black; }");

generatePermutations(0, size, array);

ui->label\_3->setText("Программа завершена.");

ui->label\_3->setStyleSheet("QLabel {color : green; }");

count=0;

inputString.clear();

}

void MainWindow::**on\_action\_3\_triggered**()

{

QMessageBox::about(this, tr("О программе"), tr("Данная программа выводит все перестановки\nзаданных N различных натуральных чисел"));

return;

}

void MainWindow::**on\_action\_4\_triggered**()

{

QMessageBox::about(this, tr("Об авторе"), tr("Лабораторная работа №1\n\nВыполнил: Ковынев М.В.\nГруппа: 6304"));

return;

}

# **ПРИЛОЖЕНИЕ В.**

# **ИХОДНЫЙ КОД 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\_action\_triggered**();

void **on\_action\_2\_triggered**();

void **on\_pushButton\_clicked**();

void **on\_pushButton\_2\_clicked**();

void **on\_action\_3\_triggered**();

void **generatePermutations**(int, int, QString\*);

void **on\_action\_4\_triggered**();

private:

Ui::MainWindow \*ui;

};

#endif // MAINWINDOW\_H