|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ **Информатика и системы управления**

КАФЕДРА **Компьютерные системы и сети (ИУ6)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.03.01 Информатика и вычислительная техника**

**Отчет**

|  |  |
| --- | --- |
| **По лабораторной работе №** | 10 |

**Название:**

Qt. Создание контейнеров

**Дисциплина:** Объектно-ориентированное программирование

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ6-22Б |  |  | И.А. Люляев |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  |  |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

Москва, 2020

Лабораторная работа №10

**Лабораторная работа 10. Qt. Создание контейнеров**

*Моделировать дек, в качестве элементов которого могут использоваться вещественные числа и строки. Операции: добавление элемента, удаление элемента, печать элементов дека. При добавлении и удалении элементов выбирать сторону дека: начало или конец. Создать класс-потомок, который содержит функцию поиска строки максимальной длины. Тестировать полученную модель.*

*Разработать собственную иерархию классов, готовые контейнеры Qt не использовать. Пользовательский интерфейс для работы с моделью реализовать на Qt. В отчете представить диаграмму классов и обосновать выбранную структуру представления данных.*

**Код программы:**

**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();

}

**Файл mainwindow.h**

#ifndef MAINWINDOW\_H

#define MAINWINDOW\_H

#include <QMainWindow>

QT\_BEGIN\_NAMESPACE

namespace **Ui** { class **MainWindow**; }

QT\_END\_NAMESPACE

class **MainWindow** : public QMainWindow

{

Q\_OBJECT

public:

**MainWindow**(QWidget \*parent = nullptr);

~***MainWindow***();

private:

Ui::MainWindow \*ui;

private slots:

void **addEl**();

void **deleteEl**();

void **printEl**();

void **onLineEditEnd**();

void **maxStr**();

};

#endif // MAINWINDOW\_H

**Файл mainwindow.cpp**

#include "mainwindow.h"

#include "ui\_mainwindow.h"

#include "deque.h"

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

: QMainWindow(parent)

, ui(new Ui::MainWindow)

{

ui->setupUi(this);

connect(ui->addButton, SIGNAL(clicked()), this, SLOT(addEl()));

connect(ui->addButton, SIGNAL(clicked()), this, SLOT(onLineEditEnd()));

connect(ui->deleteButton, SIGNAL(clicked()), this, SLOT(deleteEl()));

connect(ui->printButton, SIGNAL(clicked()), this, SLOT(printEl()));

connect(ui->lineEdit, SIGNAL(editingFinished()), this, SLOT(onLineEditEnd()));

connect(ui->maxString, SIGNAL(clicked()), this, SLOT(maxStr()));

}

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

{

delete ui;

}

DequeV2 \*deque = new DequeV2();

void MainWindow::**addEl**() {

QString input = ui->lineEdit->text();

if (ui->comboBox->currentText() == "begin") {

deque->addElementFront(input);

} else {

deque->addElementBack(input);

}

printEl();

}

void MainWindow::**deleteEl**() {

if (ui->comboBox->currentText() == "begin") {

deque->deleteElementFront();

} else {

deque->deleteElementBack();

}

printEl();

}

void MainWindow::**printEl**() {

ui->textEdit->setPlainText("");

ui->textEdit->append("Printing Deque elements: ");

for (int i = 0; i < (int)deque->doubles.size(); i++) {

QString element = QString::fromStdString(std::to\_string(deque->doubles[i]));

ui->textEdit->append(element);

}

for (int i = 0; i < (int)deque->strings.size(); i++) {

ui->textEdit->append(deque->strings[i]);

}

}

void MainWindow::**onLineEditEnd**() {

ui->lineEdit->setText("");

}

void MainWindow::**maxStr**() {

QString maxStr = deque->maxLengthString();

ui->textEdit->setPlainText(maxStr);

}

**Файл deque.h**

#ifndef DEQUE\_H

#define DEQUE\_H

#include <QString>

#include <vector>

class **Deque** {

public:

std::vector<double> doubles;

std::vector<QString> strings;

bool isDoubleAddedLast;

void **addElementBack**(QString input);

void **addElementFront**(QString input);

void **deleteElementBack**();

void **deleteElementFront**();

};

class **DequeV2** : public Deque {

public:

QString **maxLengthString**();

};

#endif // DEQUE\_H

**Файл deque.cpp**

#include "deque.h"

void Deque::**addElementBack**(QString input) {

bool isDouble = false;

double element = input.toDouble(&isDouble);

if (isDouble) {

doubles.push\_back(element);

isDoubleAddedLast = true;

} else {

strings.push\_back(input);

isDoubleAddedLast = false;

}

}

void Deque::**addElementFront**(QString input) {

bool isDouble = false;

input.toDouble(&isDouble);

if (isDouble) {

doubles.insert(doubles.begin(), input.toDouble());

isDoubleAddedLast = true;

} else {

strings.insert(strings.begin(), input);

isDoubleAddedLast = false;

}

}

void Deque::**deleteElementBack**() {

if ((doubles.size() == 0) && (strings.size() == 0)) return;

if (isDoubleAddedLast) {

if (doubles.size() != 0) doubles.pop\_back();

else isDoubleAddedLast = !isDoubleAddedLast;

} else {

if (strings.size() != 0) strings.pop\_back();

else isDoubleAddedLast = !isDoubleAddedLast;

}

}

void Deque::**deleteElementFront**() {

if ((doubles.size() == 0) && (strings.size() == 0)) return;

if (isDoubleAddedLast) {

if (doubles.size() != 0) doubles.erase(doubles.begin());

else isDoubleAddedLast = !isDoubleAddedLast;

} else {

if (strings.size() != 0) strings.erase(strings.begin());

else isDoubleAddedLast = !isDoubleAddedLast;

}

}

QString DequeV2::**maxLengthString**() {

QString maxStr = "";

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

if (strings[i] > maxStr) maxStr = strings[i];

}

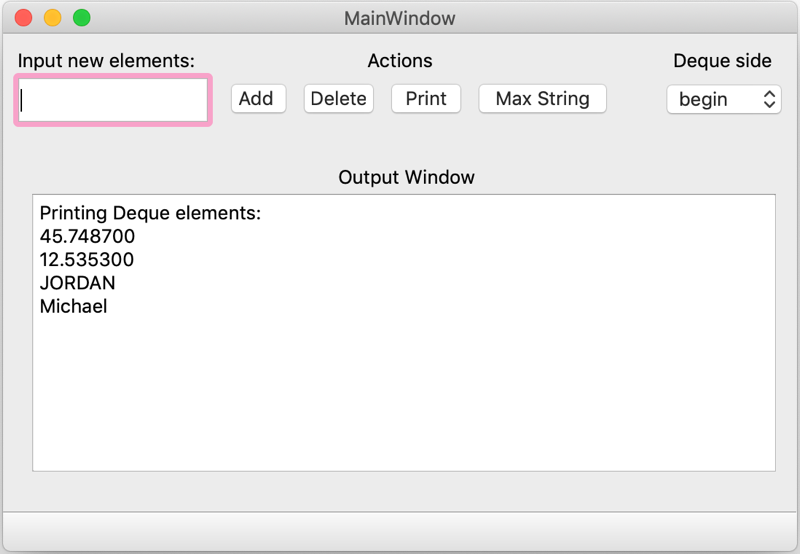
if (strings.size() == 0) maxStr = "No strings are added";

return maxStr;

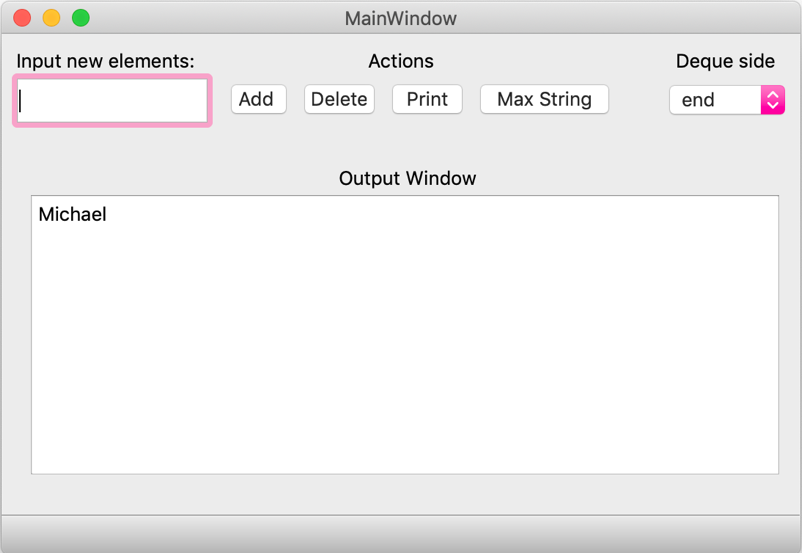
}

**Результат:**

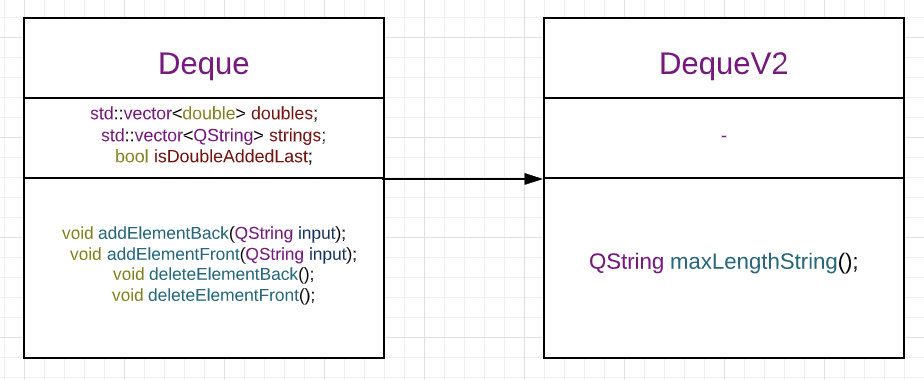
Введено 4 элемента:

****

Нажата кнопка “Max String”

****

**Диаграмма классов:**

****

Данная структура была выбрана, так как для требуемого функционала наиболее удобным является размещение всех элементов на одном окне. Также данная структура более понятна для чтения другим программистам, которые потенциально будут использовать и модифицировать эту программу.

**Вывод:**

В ходе лабораторной работы, был разработан контейнер на C++ с реализацией взаимодействия с ним через интерфейс, созданный в QT.