

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В. Г. ШУХОВА»
(БГТУ им. В.Г. Шухова)**

Кафедра программного обеспечения вычислительной техники и
автоматизированных систем

Лабораторная работа №10

по дисциплине: Объектно-ориентированное программирование

Тема: Закрепление навыков программирования в объектно-
ориентированном стиле. Визуальные компоненты. Знакомство с
QT.

Выполнил: студент группы ПВ-223

Дмитриев А.А.

Проверил:

Черников С.В.

Цель работы: приобретение практических навыков создания приложений на языке C++.

Задание:

Для выполнения лабораторных работ с использованием операционной системы Microsoft Windows необходимо установить следующее программное обеспечение: Tortoise Mercurial актуальной версии; Qt SDK или иной дистрибутив Qt согласованный с преподавателем. При изучении дисциплины необходимо обеспечить доступ в интернет для осуществления доступа к системе контроля версий. Лабораторные работы выполняются на объектно-ориентированном языке программирования C++. Изменения, вносимые студентами в свои проекты, следует периодически отдавать на сервер системы контроля версий для осуществления контроля над выполнением заданий со стороны преподавателя. Для успешной сдачи лабораторной работы следует выполнить следующий перечень заданий.

Задание 1.

Создать репозиторий под контролем git на открытой площадке gitlab.com
Задать имя репозитория <год>_<группа>_<имя студента в транслите>_<номер варианта по журналу>.

Задание 2.

Выполнить проектирование задачи в соответствии с вариантом (табл. 1).
Для реализации поставленной задачи необходимо спроектировать, реализовать и использовать шаблон «умные указатели». Соответственно это учесть при проектировании программного обеспечения.

Задача 3.

Выполнить реализацию в соответствии с вариантом задачи (табл. 1), используя среду разработки QT. Реализация должна быть кроссплатформенной и выполнена на основе графических окон.

2 вариант: Создать приложения для просмотра графического материала, поиск файлов с разрешениями jpeg, bmp, png, сохранение ссылок на них в файл, для быстрого доступа при повторном запуске приложения.

№1

```
commit 952812c238b625a9a114f5ca84377c7177306253 (HEAD -> lab_10_task_1)
Author: AnDreV133 <dmitriyikubik@gmail.com>
Date: Sat Apr 13 17:28:49 2024 +0300
```

done

```
commit ddd2c58c66f0f4a0c50c23a96015c15678e074ae
Author: AnDreV133 <dmitriyikubik@gmail.com>
Date: Sat Apr 13 08:52:13 2024 +0300
```

main functional is available

```
commit c7f6306762b07a8a5bcd14e70c989a67e9135e4f
Author: AnDreV133 <dmitriyikubik@gmail.com>
Date: Thu Apr 11 00:04:11 2024 +0300
```

bad init, cause SmartPointer

```
commit 7e1b506677ba0875f87cd343b3d353bc291a59c9 (master)
Author: AnDreV133 <dmitriyikubik@gmail.com>
Date: Wed Apr 10 18:04:47 2024 +0300
```

init

№2

```
#ifndef SMARTPOINTER_H
#define SMARTPOINTER_H

template<typename T>
class SmartPointer {
    T *obj;
public:
    SmartPointer() {}
    SmartPointer(T *obj) {
        this->obj = obj;
    }
    ~SmartPointer() {
        delete obj;
    }
    T* operator->() { return obj; }
    T& operator* () { return *obj; }
};

#endif // SMARTPOINTER_H
```

№3

```
#ifndef FILESAVER_H
#define FILESAVER_H

#include <forward_list>
#include <string>
#include <fstream>

#include <QStringList>
```

```

#include <QSet>
#include <QList>

#include "ISaveable.h"
#include "SmartPointer.hpp"

class FileSaver {
private:
    std::string recordFileName;
public:
    FileSaver(std::string recordFileName) {
        this->recordFileName = recordFileName;
    }

    void save(ISaveable &s) {
        std::ofstream recordFile(recordFileName, std::ofstream::app);
        if (recordFile.is_open()) {
            recordFile << s.onSave() << '\n';
            recordFile.close();
        } else {
            throw std::ios_base::failure("make or write to file is impossible");
        }
    }

    QStringList getPathsToImages() {
        std::ifstream recordFile(recordFileName, std::ifstream::app);
        if (!recordFile.is_open()) {
            throw std::ios_base::failure("file cant be read");
        }

        SmartPointer<QSet<QString>> set = new QSet<QString>();
        std::string buf;
        while (getline(recordFile, buf))
        {
            set->insert(QString(buf.c_str()));
        }

        recordFile.close();

        return QStringList(QList<QString>(set->begin(), set->end()));
    }
};

#endif // FILESAVER_H

#ifndef ISAVEABLE_H
#define ISAVEABLE_H

#include <string>

class ISaveable {
public:
    virtual std::string onSave() = 0;
};

#endif // ISAVEABLE_H

#ifndef MAINWINDOW_H
#define MAINWINDOW_H

```

```

#include <QMainWindow>

#include "Viewer.h"
#include "SmartPointer.hpp"
#include "mainwindow.h"
#include "ui_mainwindow.h"
#include "FileSaver.h"

namespace Ui {
class MainWindow;
}

class MainWindow : public QMainWindow
{
    Q_OBJECT

public:
    explicit MainWindow(QWidget *parent = nullptr);
    ~MainWindow();

private slots:
    void on_showButton_clicked();
    void on_reviewButton_clicked();

private:
    Ui::MainWindow *ui;
    QString fileName;
    QWidget* widget;
    FileSaver* fs;
};

#endif // MAINWINDOW_H

#include "mainwindow.h"
#include "ui_mainwindow.h"

#include <QFileDialog>
#include <QString>
#include <QLabel>
#include <QPixmap>
#include <QWindow>
#include <QWidget>

#include <memory>
#include <iostream>
#include <exception>

#include "SmartPointer.hpp"
#include "FileSaver.h"

MainWindow::MainWindow(QWidget *parent)
    : QMainWindow(parent)
    , ui(new Ui::MainWindow)
{
    ui->setupUi(this);

    fs = new FileSaver("cache.txt");
    ui->comboBox->addItem(fs->getPathsToImages());
}

```

```

MainWindow::~MainWindow()
{
    delete ui;
}

void MainWindow::on_reviewButton_clicked()
{
    fileName = QFileDialog::getOpenFileName(
        this,
        QString(),
        QString(),
        "jpeg (*.jpeg);;bmp (*.bmp);;png (*.png)"
    );

    SmartPointer<Viewer> v = new Viewer(fileName.toStdString());
    ui->comboBox->setCurrentText(fileName);

    fs->save(*v);

    ui->comboBox->clear();
    ui->comboBox->addItem(fs->getPathsToImages());
}

void MainWindow::on_showButton_clicked()
{
    widget = new QWidget();

    QLabel *label = new QLabel();
    QString path = ui->comboBox->currentText();

    qDebug(path.toStdString().c_str());

    label->setText(path);
    label->setPixmap(QPixmap(path));
    QVBoxLayout *layout = new QVBoxLayout();
    layout->addWidget(label);

    widget->setLayout(layout);
    widget->show();
}

#ifdef VIEWER_H
#define VIEWER_H

#include <string>
#include "ISaveable.h"

class Viewer : public ISaveable {
private:
    std::string filePath;
    std::string getFileExtension(const std::string& filePath) {
        auto pos = filePath.rfind('.');
        if (pos <= 0) return "";
        return filePath.substr(pos+1, std::string::npos);
    }

public:
    Viewer(const std::string &filePath) {
        std::string fileExt = getFileExtension(filePath);
        if (fileExt == "jpeg" || fileExt == "bmp" || fileExt == "png")

```

```

        this->filePath = filePath;
    else
        std::__throw_runtime_error("uncorrect file extension.");
    }

    std::string onSave() override {
        return filePath;
    }
};

```

```
#endif // VIEWER_H
```

```
#include "mainwindow.h"
```

```

#include <QApplication>
#include <QLocale>
#include <QTranslator>
#include <iostream>

```

```

int main(int argc, char *argv[])
{
    QApplication a(argc, argv);

    MainWindow w;
    w.show();

    return a.exec();
}

```

Пример работы:

```

graph TD
    Start([def output_saddle_point  
(matrix_game)]) --> Print[print("Стратегии игроков.")  
print(gm)  
res = get_saddle_point(matrix_game)]
    Print --> Decision{res is None ?}
    Decision -- Yes --> PrintYes[print("Седловая  
точка не найдена")]
    Decision -- No --> PrintNo[print("Седловая точка равна:", res.cost)  
print("Выиграл игрок:", res.player)]
    PrintYes --> End([End])
    PrintNo --> End

```

MainWindow

C:/Users/dmitri/Pictures/Screenshots/Снимок экрана 2024-03-24 165506.png

Обзор

Показать

object

Предустановки точки останова

Файл

...ainwindow.cpp

OP\lab10\build\Desktop_Qt_shared_MinGW_w64_MINGW64_MSYS2-Debug\lab10.exe...

-03-23 230306.png

-03-23 230306.png

-03-18 221705.png

-03-24 165506.png

Вывод: В ходе лабораторной работы закрепили навыки программирования в объектно-ориентированном стиле. Познакомились с инструментарием QT.