Министерство высшего образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Пермский национальный исследовательский политехнический университет» (ПНИПУ)

Электротехнический факультет

Кафедра «Информационные технологии и автоматизированные системы»

ОТЧЁТ

по работе

Тема: «ARM Репетитора»

Выполнил

Студент группы РИС-22-2б

Юсупов Т.А.

Проверил доц. Кафедры ИТАС

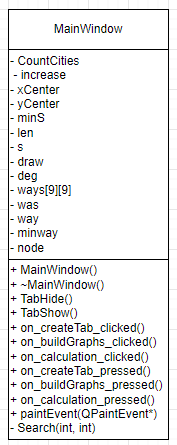
Полякова О. А.

Пермь 2023

**Постановка задачи**

1. Создание и разработка рабочего места репетитора.
2. В программе должны генерироваться две таблицы, 1 – таблица с недельным планом занятости, 2 – таблица с проведенными занятиями. Ученики и все сопутствующие данные генерируются рандомизированным образом.

**Uml-Диаграмма**



**Код программы**

**main.cpp:**

#include <QApplication>

#include "mainwindow.h"

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

QApplication app(*argc*, *argv*);

MainWindow mainWindow;

mainWindow.show();

*return* app.exec();

}

**mainwindow.h:**

#ifndef MAINWINDOW\_H

#define MAINWINDOW\_H

#include <QWidget>

#include <QPushButton>

#include <QVBoxLayout>

#include "completedlessonswindow.h"

#include "weeklyplanwindow.h"

*class* **MainWindow** : *public* QWidget {

Q\_OBJECT

*public*:

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

*private* slots:

void **openWeeklyPlan**();

void **openCompletedLessons**();

*private*:

QVBoxLayout \*mainLayout;

CompletedLessonsWindow \*completedLessonsWindow;

};

#endif *//* *MAINWINDOW\_H*

**mainwindow.cpp:**

#include "mainwindow.h"

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

: QWidget(*parent*) {

setWindowTitle("Расписание");

setGeometry(100, 100, 200, 100);

mainLayout = *new* QVBoxLayout(*this*);

QPushButton \*weeklyPlanButton = *new* QPushButton("Недельный план");

connect(weeklyPlanButton, &QPushButton::clicked, *this*, &MainWindow::openWeeklyPlan);

mainLayout->addWidget(*weeklyPlanButton*);

QPushButton \*completedLessonsButton = *new* QPushButton("Проведенные занятия");

connect(completedLessonsButton, &QPushButton::clicked, *this*, &MainWindow::openCompletedLessons);

mainLayout->addWidget(*completedLessonsButton*);

completedLessonsWindow = *new* CompletedLessonsWindow();

setLayout(*mainLayout*);

}

void MainWindow::**openWeeklyPlan**() {

WeeklyPlanWindow \*weeklyPlanWindow = *new* WeeklyPlanWindow();

weeklyPlanWindow->*exec*();

}

void MainWindow::**openCompletedLessons**() {

completedLessonsWindow->show();

}

**weeklyplanwindow.h:**

#ifndef WEEKLYPLANWINDOW\_H

#define WEEKLYPLANWINDOW\_H

#include <QDialog>

#include <QTableWidget>

#include <QHeaderView>

#include <QTableWidgetItem>

#include <QRandomGenerator>

*class* **WeeklyPlanWindow** : *public* QDialog {

Q\_OBJECT

*public*:

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

*private*:

QTableWidget \*table;

QStringList **generateRandomNames**(int count);

};

#endif *//* *WEEKLYPLANWINDOW\_H*

#include "weeklyplanwindow.h"

WeeklyPlanWindow::**WeeklyPlanWindow**(QWidget \*parent)

: QDialog(*parent*) {

setWindowTitle("Расписание");

setGeometry(100, 100, 1100, 400);

table = *new* QTableWidget(*this*);

table->setGeometry(10, 10, 1000, 385);

int rowCount = 12;

int columnCount = 6;

table->setRowCount(rowCount);

table->setColumnCount(columnCount);

QStringList headers;

headers << "Понедельник" << "Вторник" << "Среда" << "Четверг" << "Пятница" << "Суббота";

table->setHorizontalHeaderLabels(headers);

QStringList timeIntervals;

timeIntervals << "8:00-9:00" << "9:00-10:00" << "10:00-11:00" << "11:00-12:00"

<< "12:00-13:00" << "13:00-14:00" << "14:00-15:00" << "15:00-16:00"

<< "16:00-17:00" << "17.00-18.00" << "18.00-19.00" << "19.00-20.00";

table->setVerticalHeaderLabels(timeIntervals);

table->horizontalHeader()->setDefaultAlignment(Qt::*AlignHCenter*);

table->verticalHeader()->setDefaultAlignment(Qt::*AlignHCenter*);

table->setColumnWidth(0, 100);

*for* (int col = 0; col < columnCount; ++col) {

table->setColumnWidth(col, 150);

}

QStringList studentNames = generateRandomNames(rowCount \* columnCount);

QSet<int> filledCells;

*for* (int col = 1; col <= columnCount; ++col) {

int filledCellCount = QRandomGenerator::global()->bounded(2, 4);

filledCells.clear();

*while* (filledCells.size() < filledCellCount) {

int row = QRandomGenerator::global()->bounded(rowCount);

filledCells.insert(row);

}

foreach (int row, filledCells) {

QTableWidgetItem \*item = *new* QTableWidgetItem(studentNames.at(row \* columnCount + col - 1));

table->setItem(row, col, *item*);

}

}

}

QStringList WeeklyPlanWindow::**generateRandomNames**(int count) {

QStringList names;

QStringList surnames;

names << "Иван" << "Александр" << "Николай" << "Василий" << "Павел";

surnames << "Иванов" << "Петров" << "Сидоров" << "Смирнов" << "Кузнецов";

QStringList randomNames;

*for* (int i = 0; i < count; ++i) {

QString name = names.at(QRandomGenerator::global()->bounded(names.size()));

QString surname = surnames.at(QRandomGenerator::global()->bounded(surnames.size()));

randomNames << QString("%1 %2").arg(name).arg(surname);

}

*return* randomNames;

}

**completedlessonswindow.h:**

#ifndef COMPLETEDLESSONSWINDOW\_H

#define COMPLETEDLESSONSWINDOW\_H

#include <QDialog>

#include <QTableWidget>

#include <QHeaderView>

#include <QTableWidgetItem>

#include <QRandomGenerator>

#include <QSet>

#include <QPushButton>

#include <QVBoxLayout>

#include <QMessageBox>

#include <QDate>

*class* **CompletedLessonsWindow** : *public* QDialog {

Q\_OBJECT

*public*:

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

*private* slots:

void **addLesson**();

*private*:

QTableWidget \*table;

QStringList **generateRandomNames**(int count);

};

#endif *//* *COMPLETEDLESSONSWINDOW\_H*

**completedlessonswindow.cpp:**

#include "completedlessonswindow.h"

CompletedLessonsWindow::**CompletedLessonsWindow**(QWidget \*parent)

: QDialog(*parent*) {

setWindowTitle("Проведенные занятия");

setGeometry(100, 100, 600, 600);

QVBoxLayout \*mainLayout = *new* QVBoxLayout(*this*);

table = *new* QTableWidget(*this*);

table->setGeometry(10, 10, 580, 280);

int rowCount = 15;

int columnCount = 4;

table->setRowCount(rowCount);

table->setColumnCount(columnCount);

QStringList headers;

headers << "Имя" << "Дата" << "Продолжительность" << "Стоимость";

table->setHorizontalHeaderLabels(headers);

table->horizontalHeader()->setDefaultAlignment(Qt::*AlignHCenter*);

table->verticalHeader()->setDefaultAlignment(Qt::*AlignHCenter*);

table->horizontalHeader()->setSectionResizeMode(QHeaderView::*Stretch*);

table->verticalHeader()->setSectionResizeMode(QHeaderView::*Stretch*);

QStringList studentNames = generateRandomNames(rowCount);

*for* (int row = 0; row < rowCount; ++row) {

QTableWidgetItem \*itemName = *new* QTableWidgetItem(studentNames.at(row));

table->setItem(row, 0, *itemName*);

QDate date = QDate(2023, 5, 1).addDays(QRandomGenerator::global()->bounded(31));

QTableWidgetItem \*itemDate = *new* QTableWidgetItem(date.toString("yyyy.MM.dd"));

table->setItem(row, 1, *itemDate*);

QTableWidgetItem \*itemDuration = *new* QTableWidgetItem("60 мин");

table->setItem(row, 2, *itemDuration*);

QTableWidgetItem \*itemCost = *new* QTableWidgetItem("600");

table->setItem(row, 3, *itemCost*);

}

mainLayout->addWidget(*table*);

QPushButton \*addButton = *new* QPushButton("Добавить занятие");

connect(addButton, &QPushButton::clicked, *this*, &CompletedLessonsWindow::addLesson);

mainLayout->addWidget(*addButton*);

setLayout(*mainLayout*);

}

QStringList CompletedLessonsWindow::**generateRandomNames**(int count) {

QStringList names;

QStringList surnames;

names << "Иван" << "Александр" << "Николай" << "Василий" << "Павел";

surnames << "Иванов" << "Петров" << "Сидоров" << "Смирнов" << "Кузнецов";

QStringList randomNames;

*for* (int i = 0; i < count; ++i) {

QString name = names.at(QRandomGenerator::global()->bounded(names.size()));

QString surname = surnames.at(QRandomGenerator::global()->bounded(surnames.size()));

randomNames << QString("%1 %2").arg(name).arg(surname);

}

*return* randomNames;

}

void CompletedLessonsWindow::**addLesson**() {

int newRow = table->rowCount();

table->insertRow(newRow);

}

**Ключевые функции**

В коде есть следующие функции:

1. `QStringList generateRandomNames(int count)` - генерирует случайные имена и фамилии для студентов. Она принимает количество имен, которые нужно сгенерировать, и возвращает список случайных имен и фамилий.

2. `void openWeeklyPlan()` - открывает диалоговое окно с недельным планом. Создает таблицу с расписанием занятий и заполняет ее случайными именами студентов.

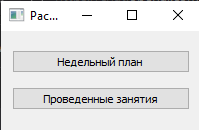
3. `void openCompletedLessons()` - открывает диалоговое окно с проведенными занятиями. Создает таблицу с данными о проведенных занятиях, включая имя студента, дату, продолжительность и стоимость занятия. Также добавляет кнопку "Добавить занятие" для добавления новой строки в таблицу.

4. `void addLesson()` - слот, вызываемый при нажатии на кнопку "Добавить занятие" в окне проведенных занятий. Добавляет новую строку в таблицу с пустыми ячейками.

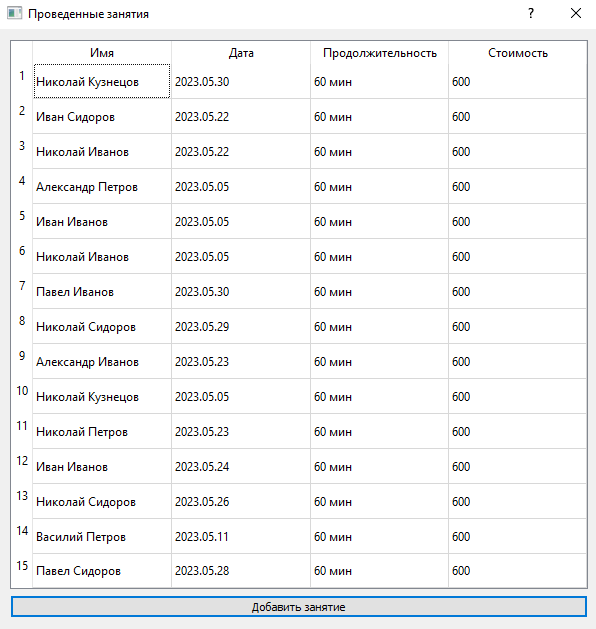
5. `int main(int argc, char \*argv[])` - главная функция программы. Создает экземпляр приложения `QApplication` и инициализирует список случайных имен студентов. Создает главное окно приложения и добавляет кнопки "Недельный план" и "Проведенные занятия". При нажатии на кнопки вызываются функции `openWeeklyPlan` и `openCompletedLessons` соответственно. Запускает главный цикл приложения.

В классе `WeeklyPlanWindow` и `CompletedLessonsWindow` добавлены конструкторы и приватные члены класса для хранения таблицы и обработки событий. Функции `generateRandomNames` и `addLesson` также были перенесены в соответствующие классы.

**Вывод**

****

****

****