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 лабораторной работе

Тема: «АРМ библиотекаря»

Выполнил

Студент группы РИС-22-2б

Арапов М.В.

Проверил доц. Кафедры ИТАС

Полякова О. А.

Пермь 2023

**Постановка задачи**

1) Все книги - массив. Один элемент состоит из автора, названия и года выпуска

2) Библиотекарь может управлять массивом. Его возможности:

* Вносить новые книги в массив
* Удалять книги
* Смотреть какие книги сейчас есть в наличии
* Смотреть какие книги выданы (тут же видеть срок, когда их должны вернуть)
* Выдавать книги
* Получать книги обратно

3) В тестовом режиме будут генерироваться новые читатели, которые будут просить одну рандомную книгу.  
  
**Функции в коде:**

**Функция isBookExists** проверяет, существует ли книга с указанным названием в файле "books.txt". Если книга существует, функция возвращает true, в противном случае - false.

**Функция writeBookToFile** записывает информацию о книге (автор и название) в файл "books.txt", если книга с указанным названием не существует. Если книга уже существует, выводится сообщение об ошибке.

**Функция removeBookFromFile** удаляет книгу с указанным названием из файла "books.txt", если она существует. Если книга не найдена, выводится сообщение об ошибке.

**Функция getAvailableBooksFromFile** возвращает список доступных книг из файла "books.txt". В списке отображаются только книги, которые не были выданы читателям.

**Функция getIssuedBooksFromFile** возвращает список выданных книг из файла "books.txt". В списке отображаются только книги, которые были выданы читателям.

**Функция borrowBook** берет книгу с указанным названием из файла "books.txt" на выдачу читателю с указанными именем, фамилией и номером телефона. Если книга не найдена или уже выдана, выводится сообщение об ошибке.

**Функция returnBook** возвращает книгу с указанным названием в библиотеку путем удаления информации о выдаче из файла "books.txt". Если книга не была выдана, выводится сообщение об ошибке.

**Функция getDelayedBooksFromFile** возвращает список книг из файла "books.txt", которые находятся на руках у читателей и просрочены по сроку возврата.

**Функция generateRandomReader** генерирует случайные данные о читателе (имя, фамилия, номер телефона) и случайную желаемую книгу для выдачи. Возвращает сгенерированные данные в виде строки.

**Функция generateRandomReaders** генерирует заданное количество случайных читателей с помощью функции generateRandomReader. Возвращает список сгенерированных данных о читателях.

**Код программы**

#include <QtWidgets>

#include <QString>

#include <QDateTime>

#include <iostream>

#include <QFile>

#include <QApplication>

*using* *namespace* std;

bool **isBookExists**(*const* QString& title) {

QFile file("books.txt");

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

*while* (!file.*atEnd*()) {

QString line = file.readLine();

QStringList bookInfo = line.split("-");

*if* (bookInfo.size() == 2 && bookInfo[1].trimmed() == title) {

file.*close*();

*return* *true*;

}

}

file.*close*();

}

*return* *false*;

}

void **writeBookToFile**(*const* QString& author, *const* QString& title) {

*if* (isBookExists(title)) {

QMessageBox::critical(*nullptr*, "Ошибка", "Книга уже существует");

*return*;

}

QFile file("books.txt");

*if* (file.*open*(QIODevice::*WriteOnly* | QIODevice::*Append* | QIODevice::*Text*)) {

QTextStream stream(*&file*);

stream.setCodec("UTF-8");

QString bookData = author + "-" + title;

stream << bookData << "\n";

file.*close*();

}

}

void **removeBookFromFile**(*const* QString& title) {

*if* (!isBookExists(title)) {

QMessageBox::critical(*nullptr*, "Ошибка", "Книга не найдена");

*return*;

}

QFile inputFile("books.txt");

*if* (inputFile.*open*(QIODevice::*ReadOnly*)) {

QStringList bookData;

bool bookFound = *false*;

*while* (!inputFile.*atEnd*()) {

QString line = inputFile.readLine();

QStringList bookInfo = line.split("-");

*if* (bookInfo.size() == 2 && bookInfo[1].trimmed() == title) {

bookFound = *true*;

} *else* {

bookData << line.trimmed();

}

}

inputFile.*close*();

*if* (!bookFound) {

QMessageBox::critical(*nullptr*, "Ошибка", "Такой книги нет");

*return*;

}

QFile outputFile("books.txt");

*if* (outputFile.*open*(QIODevice::*WriteOnly* | QIODevice::*Truncate* | QIODevice::*Text*)) {

QTextStream stream(*&outputFile*);

*for* (*const* QString& line : bookData) {

stream << line << "\n";

stream.setCodec("UTF-8");

}

outputFile.*close*();

}

}

}

QStringList **getAvailableBooksFromFile**() {

QFile file("books.txt");

QStringList availableBooks;

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

*while* (!file.*atEnd*()) {

QString line = file.readLine().trimmed();

QStringList bookInfo = line.split("-");

*if* (bookInfo.size() == 2 && !line.contains("(")) {

QString bookData = line.trimmed();

availableBooks << bookData;

}

}

file.*close*();

}

*return* availableBooks;

}

QStringList **getIssuedBooksFromFile**() {

QFile file("books.txt");

QStringList issuedBooks;

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

*while* (!file.*atEnd*()) {

QString line = file.readLine().trimmed();

QStringList bookInfo = line.split("-");

*if* (bookInfo.size() > 2 && line.contains("(")) {

QString issuedBook = line.trimmed();

issuedBooks << issuedBook;

}

}

file.*close*();

}

*return* issuedBooks;

}

void **borrowBook**(*const* QString& title, *const* QString& readerName, *const* QString& readerSurname, *const* QString& phoneNumber) {

*if* (!isBookExists(title)) {

QMessageBox::critical(*nullptr*, "Ошибка", "Книга не найдена");

*return*;

}

QFile inputFile("books.txt");

*if* (inputFile.*open*(QIODevice::*ReadOnly*)) {

QStringList bookData;

bool bookFound = *false*;

*while* (!inputFile.*atEnd*()) {

QString line = inputFile.readLine();

QStringList bookInfo = line.split("-");

*if* (bookInfo.size() == 2 && bookInfo[1].trimmed() == title) {

*if* (bookInfo.size() > 2) {

QMessageBox::critical(*nullptr*, "Ошибка", "Книга уже выдана");

inputFile.*close*();

*return*;

}

QDateTime currentDate = QDateTime::currentDateTime();

QDateTime dueDate = currentDate.addSecs(120); *//* *Добавление* *2* *минут* *(120* *секунд)*

QString formattedDate = dueDate.toString("yyyy.MM.dd hh:mm:ss");

line = bookInfo[0].trimmed() + "-" + bookInfo[1].trimmed() + "-" + "(" + readerName + " " + readerSurname + " " + phoneNumber + "-" + formattedDate + ")";

qDebug() << "Due Date:" << dueDate;

bookFound = *true*;

}

bookData << line.trimmed();

}

inputFile.*close*();

*if* (!bookFound) {

QMessageBox::critical(*nullptr*, "Ошибка", "Такой книги нет");

*return*;

}

QFile outputFile("books.txt");

*if* (outputFile.*open*(QIODevice::*WriteOnly* | QIODevice::*Truncate* | QIODevice::*Text*)) {

QTextStream stream(*&outputFile*);

stream.setCodec("UTF-8");

*for* (*const* QString& line : bookData) {

stream << line << "\n";

}

outputFile.*close*();

}

}

}

void **returnBook**(*const* QString& title) {

QFile inputFile("books.txt");

*if* (inputFile.*open*(QIODevice::*ReadOnly*)) {

QStringList bookData;

bool bookFound = *false*;

*while* (!inputFile.*atEnd*()) {

QString line = inputFile.readLine();

QStringList bookInfo = line.split("-");

*if* (bookInfo.size() >2 && bookInfo[1].trimmed() == title) {

*for* (int i = 0; i < 2; ++i) {

bookInfo.removeLast();

}

line = bookInfo.join("-") + "\n";

bookFound = *true*;

}

bookData << line.trimmed();

}

inputFile.*close*();

*if* (!bookFound) {

QMessageBox::critical(*nullptr*, "Ошибка", "Такая книга не была выдана");

*return*;

}

QFile outputFile("books.txt");

*if* (outputFile.*open*(QIODevice::*WriteOnly* | QIODevice::*Truncate* | QIODevice::*Text*)) {

QTextStream stream(*&outputFile*);

stream.setCodec("UTF-8");

*for* (*const* QString& line : bookData) {

stream << line << "\n";

}

outputFile.*close*();

}

}

}

QStringList **getDelayedBooksFromFile**() {

QFile file("books.txt");

QStringList delayedBooks;

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

*while* (!file.*atEnd*()) {

QString line = file.readLine().trimmed();

QStringList bookInfo = line.split("-");

*if* (bookInfo.size() > 2 && line.contains("(")) {

QString dueDateStr = bookInfo.last().trimmed();

qDebug() << "Due Date Str: " << dueDateStr;

*//* *Удаление* *символа* *")"*

dueDateStr.remove(")");

qDebug() << "Modified Due Date Str: " << dueDateStr;

QDateTime dueDate = QDateTime::fromString(dueDateStr, "yyyy.MM.dd hh:mm:ss");

qDebug() << "Due Date:" << dueDate;

*if* (QDateTime::currentDateTime() > dueDate) {

QString delayedBook = line.trimmed();

delayedBooks << delayedBook;

}

}

}

file.*close*();

}

*return* delayedBooks;

}

QString **generateRandomReader**() {

*const* QStringList names = {"Иван", "Алексей", "Олег", "Тихомир"};

*const* QStringList surnames = {"Иванов", "Петров", "Сидоров", "Смирнов"};

*const* QStringList phoneNumbers = {"111-111-111", "222-222-222", "333-333-333", "444-444-444"};

qsrand(QDateTime::currentMSecsSinceEpoch());

*const* QString name = names[qrand() % names.size()];

*const* QString surname = surnames[qrand() % surnames.size()];

*const* QString phoneNumber = phoneNumbers[qrand() % phoneNumbers.size()];

QString reader = name + " " + surname + " - " + phoneNumber;

QStringList availableBooks = getAvailableBooksFromFile();

*if* (!availableBooks.isEmpty()) {

int randomIndex = qrand() % availableBooks.size();

QString randomBook = availableBooks[randomIndex];

reader += "\nХочет взять книгу: " + randomBook;

}

*return* reader;

}

QStringList **generateRandomReaders**(int count) {

QStringList readers;

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

readers << generateRandomReader();

}

*return* readers;

}

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

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

QWidget mainWidget;

mainWidget.resize(1920, 1080);

QVBoxLayout\* layout = *new* QVBoxLayout();

layout->setContentsMargins(0, 0, 0, 0); *//* *Удаление* *отступов* *вокруг* *макета*

layout->setSpacing(0); *//* *Удаление* *промежутков* *между* *элементами* *в* *макете*

mainWidget.setLayout(*layout*);

QPalette pal = mainWidget.palette();

pal.setBrush(QPalette::*Background*, QBrush(QPixmap("C:/Users/OLEG/Desktop/APM/lib.jpg")));

mainWidget.setPalette(pal);

mainWidget.setAutoFillBackground(*true*);

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

QPushButton\* removeButton = *new* QPushButton("Удалить книгу");

QPushButton\* borrowButton = *new* QPushButton("Выдать книгу");

QPushButton\* returnButton = *new* QPushButton("Вернуть книгу");

QPushButton \*generateRandomReaderButton = *new* QPushButton("Случайный читатель");

QPushButton\* availableBooksButton = *new* QPushButton("Показать доступные книги");

QPushButton\* issuedBooksButton = *new* QPushButton("Показать выданные книги");

QPushButton\* delayedBooksButton = *new* QPushButton("Показать задерживающиеся книги");

layout->addWidget(*addButton*);

addButton->setFixedSize(1920, 135);

layout->addWidget(*removeButton*);

removeButton->setFixedSize(1920, 135);

layout->addWidget(*availableBooksButton*);

availableBooksButton->setFixedSize(1920, 135);

layout->addWidget(*issuedBooksButton*);

issuedBooksButton->setFixedSize(1920, 135);

layout->addWidget(*borrowButton*);

borrowButton->setFixedSize(1920, 135);

layout->addWidget(*returnButton*);

returnButton->setFixedSize(1920, 135);

layout->addWidget(*delayedBooksButton*);

delayedBooksButton->setFixedSize(1920, 135);

layout->addWidget(*generateRandomReaderButton*);

generateRandomReaderButton->setFixedSize(1920, 135);

*//* *Изменение* *цвета* *и* *размера* *текста* *на* *кнопках*

QString buttonStyle = "QPushButton { color: white; font-size: 50px;}";

QPalette buttonPalette = addButton->palette();

buttonPalette.setColor(QPalette::*Button*, Qt::*transparent*);

addButton->setFlat(*true*);

addButton->setPalette(buttonPalette);

addButton->setStyleSheet(buttonStyle);

removeButton->setFlat(*true*);

removeButton->setPalette(buttonPalette);

removeButton->setStyleSheet(buttonStyle);

borrowButton->setFlat(*true*);

borrowButton->setPalette(buttonPalette);

borrowButton->setStyleSheet(buttonStyle);

returnButton->setFlat(*true*);

returnButton->setPalette(buttonPalette);

returnButton->setStyleSheet(buttonStyle);

generateRandomReaderButton->setFlat(*true*);

generateRandomReaderButton->setPalette(buttonPalette);

generateRandomReaderButton->setStyleSheet(buttonStyle);

availableBooksButton->setFlat(*true*);

availableBooksButton->setPalette(buttonPalette);

availableBooksButton->setStyleSheet(buttonStyle);

issuedBooksButton->setFlat(*true*);

issuedBooksButton->setPalette(buttonPalette);

issuedBooksButton->setStyleSheet(buttonStyle);

delayedBooksButton->setFlat(*true*);

delayedBooksButton->setPalette(buttonPalette);

delayedBooksButton->setStyleSheet(buttonStyle);

*//* *Установить* *прозрачный* *фон* *кнопок*

addButton->setAttribute(Qt::*WA\_TranslucentBackground*);

removeButton->setAttribute(Qt::*WA\_TranslucentBackground*);

borrowButton->setAttribute(Qt::*WA\_TranslucentBackground*);

returnButton->setAttribute(Qt::*WA\_TranslucentBackground*);

generateRandomReaderButton->setAttribute(Qt::*WA\_TranslucentBackground*);

availableBooksButton->setAttribute(Qt::*WA\_TranslucentBackground*);

issuedBooksButton->setAttribute(Qt::*WA\_TranslucentBackground*);

delayedBooksButton->setAttribute(Qt::*WA\_TranslucentBackground*);

QObject::connect(addButton, &QPushButton::clicked, [&]() {

QDialog dialog(*&mainWidget*);

dialog.resize(400, 150);

QVBoxLayout dialogLayout(*&dialog*);

QLabel authorLabel("Автор:");

authorLabel.setFont(QFont("Arial", 14)); *//* *Установка* *нового* *размера* *шрифта*

QLineEdit authorLineEdit;

authorLineEdit.setMaximumHeight(50); *//* *Установка* *новой* *высоты* *строки* *ввода*

authorLineEdit.setFont(QFont("Arial", 14));

QLabel titleLabel("Название:");

titleLabel.setFont(QFont("Arial", 14)); *//* *Установка* *нового* *размера* *шрифта*

QLineEdit titleLineEdit;

titleLineEdit.setMaximumHeight(50); *//* *Установка* *новой* *высоты* *строки* *ввода*

titleLineEdit.setFont(QFont("Arial", 14));

QDialogButtonBox buttonBox(QDialogButtonBox::*Ok* | QDialogButtonBox::*Cancel*);

dialogLayout.addWidget(*&authorLabel*);

dialogLayout.addWidget(*&authorLineEdit*);

dialogLayout.addWidget(*&titleLabel*);

dialogLayout.addWidget(*&titleLineEdit*);

dialogLayout.addWidget(*&buttonBox*);

QObject::connect(&buttonBox, &QDialogButtonBox::accepted, [&]() {

*const* QString author = authorLineEdit.text().trimmed();

*const* QString title = titleLineEdit.text().trimmed();

writeBookToFile(author, title);

dialog.close();

});

QObject::connect(&buttonBox, &QDialogButtonBox::rejected, [&]() {

dialog.close();

});

dialog.setLayout(*&dialogLayout*);

dialog.*exec*();

});

QObject::connect(removeButton, &QPushButton::clicked, [&]() {

QDialog dialog(*&mainWidget*);

dialog.resize(400, 150);

QVBoxLayout dialogLayout(*&dialog*);

QLabel titleLabel("Название книги:");

titleLabel.setFont(QFont("Arial", 14)); *//* *Установка* *нового* *размера* *шрифта*

QLineEdit titleLineEdit;

titleLineEdit.setMaximumHeight(50); *//* *Установка* *новой* *высоты* *строки* *ввода*

titleLineEdit.setFont(QFont("Arial", 14));

QDialogButtonBox buttonBox(QDialogButtonBox::*Ok* | QDialogButtonBox::*Cancel*);

dialogLayout.addWidget(*&titleLabel*);

dialogLayout.addWidget(*&titleLineEdit*);

dialogLayout.addWidget(*&buttonBox*);

QObject::connect(&buttonBox, &QDialogButtonBox::accepted, [&]() {

*const* QString title = titleLineEdit.text().trimmed();

removeBookFromFile(title);

dialog.close();

});

QObject::connect(&buttonBox, &QDialogButtonBox::rejected, [&]() {

dialog.close();

});

dialog.setLayout(*&dialogLayout*);

dialog.*exec*();

});

QObject::connect(availableBooksButton, &QPushButton::clicked, [&]() {

QStringList availableBooks = getAvailableBooksFromFile();

QString message = "Доступные книги:\n";

*if* (availableBooks.isEmpty()) {

message += "Нет доступных книг";

} *else* {

message += availableBooks.join("\n");

}

QDialog dialog;

QVBoxLayout layout(*&dialog*);

QTextBrowser textBrowser;

textBrowser.setText(message);

textBrowser.setFont(QFont("Arial", 14));

textBrowser.setReadOnly(*true*);

layout.addWidget(*&textBrowser*);

dialog.setWindowTitle("Доступные книги");

dialog.resize(1920, 1080);

dialog.*exec*();

});

QObject::connect(issuedBooksButton, &QPushButton::clicked, [&]() {

QStringList issuedBooks = getIssuedBooksFromFile();

QString message = "Выданные книги:\n";

*if* (issuedBooks.isEmpty()) {

message += "Нет выданных книг";

} *else* {

message += issuedBooks.join("\n");

}

QDialog dialog;

QVBoxLayout layout(*&dialog*);

QTextBrowser textBrowser;

textBrowser.setText(message);

textBrowser.setFont(QFont("Arial", 14));

textBrowser.setReadOnly(*true*);

layout.addWidget(*&textBrowser*);

dialog.setWindowTitle("Выданные книги");

dialog.resize(1920, 1080);

dialog.*exec*();

});

QObject::connect(delayedBooksButton, &QPushButton::clicked, [&]() {

QStringList delayedBooks = getDelayedBooksFromFile();

QString message = "Задерживающиеся книги:\n";

*if* (delayedBooks.isEmpty()) {

message += "Нет задерживающихся книг";

} *else* {

message += delayedBooks.join("\n");

}

QDialog dialog;

QVBoxLayout layout(*&dialog*);

QTextBrowser textBrowser;

textBrowser.setText(message);

textBrowser.setFont(QFont("Arial", 14));

textBrowser.setReadOnly(*true*);

layout.addWidget(*&textBrowser*);

dialog.setWindowTitle("Задерживающиеся книги");

dialog.resize(1920, 1080);

dialog.*exec*();

});

QObject::connect(borrowButton, &QPushButton::clicked, [&]() {

QDialog dialog(*&mainWidget*);

dialog.resize(500, 300);

QVBoxLayout dialogLayout(*&dialog*);

QLabel titleLabel("Название книги:");

titleLabel.setFont(QFont("Arial", 14)); *//* *Установка* *нового* *размера* *шрифта*

QLineEdit titleLineEdit;

titleLineEdit.setMaximumHeight(50); *//* *Установка* *новой* *высоты* *строки* *ввода*

titleLineEdit.setFont(QFont("Arial", 14));

QLabel readerNameLabel("Имя читателя:");

readerNameLabel.setFont(QFont("Arial", 14)); *//* *Установка* *нового* *размера* *шрифта*

QLineEdit readerNameLineEdit;

readerNameLineEdit.setMaximumHeight(50); *//* *Установка* *новой* *высоты* *строки* *ввода*

readerNameLineEdit.setFont(QFont("Arial", 14));

QLabel readerSurnameLabel("Фамилия читателя:");

readerSurnameLabel.setFont(QFont("Arial", 14)); *//* *Установка* *нового* *размера* *шрифта*

QLineEdit readerSurnameLineEdit;

readerSurnameLineEdit.setMaximumHeight(50); *//* *Установка* *новой* *высоты* *строки* *ввода*

readerSurnameLineEdit.setFont(QFont("Arial", 14));

QLabel phoneNumberLabel("Номер телефона читателя:");

phoneNumberLabel.setFont(QFont("Arial", 14)); *//* *Установка* *нового* *размера* *шрифта*

QLineEdit phoneNumberLineEdit;

phoneNumberLineEdit.setMaximumHeight(50); *//* *Установка* *новой* *высоты* *строки* *ввода*

phoneNumberLineEdit.setFont(QFont("Arial", 14));

QDialogButtonBox buttonBox(QDialogButtonBox::*Ok* | QDialogButtonBox::*Cancel*);

dialogLayout.addWidget(*&titleLabel*);

dialogLayout.addWidget(*&titleLineEdit*);

dialogLayout.addWidget(*&readerNameLabel*);

dialogLayout.addWidget(*&readerNameLineEdit*);

dialogLayout.addWidget(*&readerSurnameLabel*);

dialogLayout.addWidget(*&readerSurnameLineEdit*);

dialogLayout.addWidget(*&phoneNumberLabel*);

dialogLayout.addWidget(*&phoneNumberLineEdit*);

dialogLayout.addWidget(*&buttonBox*);

QObject::connect(&buttonBox, &QDialogButtonBox::accepted, [&]() {

*const* QString title = titleLineEdit.text().trimmed();

*const* QString readerName = readerNameLineEdit.text().trimmed();

*const* QString readerSurname = readerSurnameLineEdit.text().trimmed();

*const* QString phoneNumber = phoneNumberLineEdit.text().trimmed();

borrowBook(title, readerName, readerSurname, phoneNumber);

dialog.close();

});

QObject::connect(&buttonBox, &QDialogButtonBox::rejected, [&]() {

dialog.close();

});

dialog.setLayout(*&dialogLayout*);

dialog.*exec*();

});

QObject::connect(returnButton, &QPushButton::clicked, [&]() {

QDialog dialog(*&mainWidget*);

dialog.resize(400, 150);

QVBoxLayout dialogLayout(*&dialog*);

QLabel titleLabel("Название книги:");

titleLabel.setFont(QFont("Arial", 14)); *//* *Установка* *нового* *размера* *шрифта*

QLineEdit titleLineEdit;

titleLineEdit.setMaximumHeight(50); *//* *Установка* *новой* *высоты* *строки* *ввода*

titleLineEdit.setFont(QFont("Arial", 14));

QDialogButtonBox buttonBox(QDialogButtonBox::*Ok* | QDialogButtonBox::*Cancel*);

dialogLayout.addWidget(*&titleLabel*);

dialogLayout.addWidget(*&titleLineEdit*);

dialogLayout.addWidget(*&buttonBox*);

QObject::connect(&buttonBox, &QDialogButtonBox::accepted, [&]() {

*const* QString title = titleLineEdit.text().trimmed();

returnBook(title);

dialog.close();

});

QObject::connect(&buttonBox, &QDialogButtonBox::rejected, [&]() {

dialog.close();

});

dialog.setLayout(*&dialogLayout*);

dialog.*exec*();

});

QObject::connect(generateRandomReaderButton, &QPushButton::clicked, [&]() {

QString reader = generateRandomReader();

QMessageBox msgBox;

msgBox.setText(reader);

msgBox.setWindowTitle("Случайный читатель");

*//* *Установка* *параметров* *шрифта* *и* *размера* *текста*

QFont font("Arial", 16); *//* *Здесь* *можно* *указать* *нужный* *шрифт* *и* *размер*

msgBox.setFont(font);

msgBox.*exec*();

});

QPushButton\* exitButton = *new* QPushButton("Выход");

layout->addWidget(*exitButton*);

exitButton->setFixedSize(1920, 135);

exitButton->setFlat(*true*);

exitButton->setPalette(buttonPalette);

exitButton->setStyleSheet(buttonStyle);

exitButton->setAttribute(Qt::*WA\_TranslucentBackground*);

QObject::connect(exitButton, &QPushButton::clicked, [&]() {

QApplication::quit();

});

mainWidget.setLayout(*layout*);

mainWidget.showFullScreen();

*return* app.exec();

}

**Вывод**

