

ПЕТРОЗАВОДСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИНСТИТУТ МАТЕМАТИКИ И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ  
КАФЕДРА ИНФОРМАТИКИ И МАТЕМАТИЧЕСКОГО ОБЕСПЕЧЕНИЯ

Направление подготовки бакалавриата  
01.03.02 — Прикладная математика и информатика

Отчет по практике

РАЗРАБОТКА ПРИЛОЖЕНИЯ ДЛЯ ТРЕНИРОВКИ УСТНОГО  
СЧЕТА

Выполнил:

студента 1 курса группы 22103

Никита Сергеевич Самосюк \_\_\_\_\_  
*подпись*

Петрозаводск — 2020

# Содержание

Введение	3
1 Требования к приложению	4
2 Проектирование приложения	5
3 Реализация приложения	7
Заключение	10
Приложение А. Код инициализации основного окна программы	11
Приложение Б. Код создания основного игрового виджета	12

# Введение

Цель проекта:

Разработка приложения с графическим интерфейсом пользователя, позволяющего отрабатывать навыки быстрого счета в уме.

Задачи проекта:

1. разработка игрового модуля, оценивающего количество арифметических выражений, вычисленных пользователем за 1 минуту;
2. разработка игрового модуля, оценивающего способность пользователя вычислить выражение за 10 секунд;
3. разработка модуля сохранения лучшего результата для каждого режима игры.

# 1 Требования к приложению

Следует заметить, что задача тренировки устного счета может сама по себе иметь несколько разных формулировок. Можно, например, вычислить максимальное число выражений за достаточное время, порядка одной минуты. В то же самое время можно оценивать способность пользователя вычислить выражение за короткое время - например, 10 секунд, то есть в более быстром темпе. Также следует предусмотреть возможность регулировки сложности выражений. Таким образом, при разработке приложения следует обеспечить возможность его работы в нескольких игровых режимах с возможностью сохранения результата для каждого из них.

С учетом изложенного, к разрабатываемому приложению предъявляются следующие требования:

1. приложение должно предлагать пользователю решать математические выражения в течении определенного времени;
2. приложение должно предоставлять возможность выбора уровня сложности выражений;
3. приложение должно предоставлять специальный режим, жестко ограничивающий время решения каждой задачи;
4. приложение должно сохранять лучший результат для каждого режима игры;
5. приложение должно иметь оконный графический интерфейс пользователя;
6. приложение должно работать под управлением операционных систем Windows и Linux.

## 2 Проектирование приложения

Как отмечено ранее, приложение должно предоставлять пользователю возможность выбирать между различными вариантами игры. Всего возможны 3 варианта:

1. Легкий - игрок должен в течении минуты ввести правильные ответы на как можно большее число математических выражений, имеющих вид умножения двух двузначных чисел или сложения трех двузначных чисел. При вводе пользователем правильного ответа его счет увеличивается на единицу, и выражение заменяется другим. При вводе неправильного ответа предлагается попробовать еще раз.
2. Средний - аналогичен легкому, но предлагаемые выражения более сложны, предлагается складывать и вычитать три трехзначных числа.
3. Сложный - общее время игры неограничено, но на решение каждой задачи дается не более 10 секунд. Игра продолжается до первого неверного ответа или ответа, не введенного за 10 секунд.

Приложение было спроектировано в объектно-ориентированном стиле и содержит следующие модули (классы):

1. класс основного окна приложения;
2. класс вкладки основного окна приложения, содержащий диалог выбора опций для начала игры, включая возможность ввода игроком своего имени и выбора режима игры
3. класс вкладки основного окна приложения, на которой собственно ведется игра, в которой пользователь должен за заданное время ввести ответы на как можно большее число арифметических задач;
4. класс вкладки основного окна приложения для специального игрового режима, являющийся потомком класса игровой вкладки и имеющий аналогичные интерфейсные функции, но отличающийся внутренней реализацией, ограничивая время решения каждой задачи, а не общее время игры;
5. класс вкладки основного окна приложения, содержащей список лучших результатов для каждого режима игры;

6. класс хранилища лучших результатов для каждого режима игры, отвечающий за хранение этих результатов в том числе между сеансами работы приложения.

Основное окно приложения содержит функции переключения между различными вкладками в зависимости от действий пользователя в текущей вкладке. При запуске программы отображается вкладка с приглашением ввести имя пользователя и выбрать режим игры. Также на этой вкладке находится кнопка просмотра лучших прошлых результатов для каждого игрового режима.

При нажатии на эту кнопку происходит переключение на вкладку результатов игры, а при нажатии кнопки закрытия на ней - возврат к первой вкладке.

При нажатии на кнопку начала игры создается третья вкладка, содержащая математическое выражение, поле для ввода его результата и поле с отображением таймера. На этой вкладке происходит собственно игра. По окончании игры ее результат сохраняется в списке лучших результатов, если он является лучшим, после чего активной становится вкладка с лучшими результатами.

### 3 Реализация приложения

Приложение разработано на языке C++ с использованием библиотеки Qt для создания кроссплатформенного графического интерфейса пользователя.

Основное окно приложения содержит в себе три вкладки, из которых в каждый момент времени видна только одна. Две из этих вкладок существуют постоянно. Первая из них содержит поле ввода имени пользователя, поле выбора режима игры и кнопку начала игры. Вторая постоянно существующая вкладка содержит таблицу с лучшим результатом для каждого режима игры. Одной из возможностей Qt является возможность отслеживания элементом пользовательского интерфейса (называемом виджетом в терминологии Qt) автоматически реагировать на каждое изменение состояния другого объекта, в данном случае - объекта-хранилища лучших результатов. Эта возможность позволяет виджету отображения рейтинга автоматически обновляться при добавлении в хранилище новых результатов.

Третья вкладка содержит интерфейс самой игры - поле для вывода на экран математического выражения и поля для ввода игроком результата этого выражения. Эта вкладка создается динамически при каждом начале игры и уничтожается в конце игры, так как ее поведение зависит от выбора режима игры. Эта вкладка может быть объектом одного из двух классов - для простого и среднего уровня сложности используется один класс виджета, а для сложного уровня - другой класс виджета, производный от него, в котором переопределена значительная часть поведения.

Приведем для примера код объявления класса основного окна приложения.

```
/**
 * @brief Основное окно программы
 */
class GameWindow: public QWidget
{
    Q_OBJECT

public:
    /* Создать основное окно */
    GameWindow(QWidget *parent = nullptr);
```

```

public slots:
    /* Показать вкладку лучших результатов */
    void showRating();

    /* Скрыть вкладку лучших результатов */
    void hideRating();

    /* Начать игру с выбранным именем и режимом */
    void startGame(const QString & name, const QString & mode);

    /* Закончить игру */
    void stopGame();

private:
    /* Создать основной игровой виджет */
    PlayWidget * createPlayWidget(const QString & name, const QString & mode);

private:
    /* Набор виджетов, из которых видимым будет только один */
    QStackedWidget * mainStack;

    /* Приглашение ввести имя и выбрать режим игры */
    IntroWidget * introWidget;

    /* Вкладка с лучшими результатами по режимам игры */
    RatingWidget * ratingWidget;

    /* Вкладка, на которой проходит собственно игра */
    PlayWidget * playWidget;

    /* Хранилище лучших результатов */
    RatingObject * ratingObject;
};

```

Обычно при использовании Qt основное окно приложения является объектом класса, производного от QMainWindow, но для приложения, не имеющего меню, панели инстру-



ментов и строки состояния можно использовать QWidget. Qt автоматически поместит виджет не имеющий родительского, в отдельное окно. Макрос Q\_OBJECT объявляет внутри класса служебные структуры Qt, позволяющие одним элементам управления реагировать на события в других. Команда public slots является командой Qt, означающей, что перечисленные после нее методы объекта могут быть вызваны через особый механизм связывания методов с событиями, возникающими в других объектах. Благодаря этому механизму метод showRating, делающий активной вкладку с лучшими результатами игроков, может быть вызван вкладкой с приглашением ввести имя пользователя при нажатии на ней кнопки просмотра рейтинга, без необходимости как-то учитывать наличие этой кнопки в коде объекта основного окна. Аналогично, метод hideRating, делающий основной вкладку с приглашением ввести имя пользователя и начать игру, будет вызван при нажатии кнопки «ОК» на вкладке с лучшими результатами игры.

Объект QStackedWidget - это встроенный в Qt объект переключателя между вкладками. Он определяет область в окне (в данном случае - все окно), в которой размещается несколько элементов управления, из которых в каждый момент времени виден только один. Объект ratingObject является хранилищем пользовательских результатов, включающим их чтение с диска при создании объекта, запись на диск при его уничтожении и возможность добавить новый результат с заменой существующего в случае, если новый лучше.

Больше примеров фрагментов исходного кода находится в приложениях А и Б. Вызов connect, часто встречающийся в приложении А, является вызовом специального связывающего механизма Qt, который позволяет при наступлении определенного события в одном объекте (например, нажатии на кнопку или нажатии на Enter на поле ввода) вызвать метод другого объекта (например, обработчика нажатия на кнопку в ее родительском окне). Приложение Б содержит код создания основного игрового виджета. Здесь видно, что в зависимости от выбранного пользователем режима игры создаются объекты разных классов, что позволяет добавлять новые режимы игры в виде новых подклассов класса PlayWidget, в том числе существенно отличающихся от родительского класса.

## Заключение

Было разработано приложение, предназначенное для тренировки устного счета в различных режимах. Приложение может работать в трех разных режимах:

1. простой режим, предлагающий решить максимальное число арифметических задач в течении минуты;
2. средний режим, отличающийся более сложными выражениями по сравнению с простым;
3. сложный режим, дающий пользователю не более 10 секунд на каждую задачу без права ошибиться.

Приложение может сохранять лучший из достигнутых до сих пор результатов для каждого режима игры. Программа имеет оконный графический интерфейс и работает в операционных системах Windows и Linux.

Одним из возможных направлений дальнейшей доработки приложения является его интернационализация. В данный момент интерфейс приложения реализован на английском языке, что является хорошей практикой для приложений, использующих Qt. Встроенные в эту библиотеку механизмы интернационализации позволяют легко добавлять в программу переводы интерфейса на разные языки без необходимости модификации остального исходного кода или сопровождения отдельной версии для каждого языка.

## Приложение А. Код инициализации основного окна программы

```
/**
 * @brief Инициализация основного окна программы
 * @param parent Родительский виджет, обычно nullptr
 */
GameWindow::GameWindow(QWidget *parent):
    QWidget(parent)
{
    /* Заголовок окна, минимальный размер */
    setWindowTitle(tr("Math game"));
    setMinimumSize(640, 480);

    /* Загружаем из файла конфигурации лучшие результаты */
    ratingObject = new RatingObject(this);

    /* Создаем вкладки для начала игры и лучших результатов */
    mainStack = new QStackedWidget(this);
    introWidget = new IntroWidget(this);
    ratingWidget = new RatingWidget(ratingObject, this);

    /* Реакция на нажатие кнопок начала игры, вызова лучших
     * результатов и возврата в меню
     */
    connect(
        introWidget, SIGNAL(ratingButtonPressed()),
        this, SLOT(showRating()));
    connect(
        ratingWidget, SIGNAL(okPressed()),
        this, SLOT(hideRating()));
    connect(
        introWidget, SIGNAL(startButtonPressed(QString, QString)),
```

```

        this, SLOT(startGame(QString, QString)));

    /* Добавляем вкладки начала игры и лучших результатов в
       * переключатель вкладок
       */
    mainStack->addWidget(introWidget);
    mainStack->addWidget(ratingWidget);
    mainStack->setCurrentWidget(introWidget);

    /* Разворачиваем текущую вкладку на все окно */
    QHBoxLayout * la = new QHBoxLayout();

    la->addWidget(mainStack);

    setLayout(la);
}

```

## Приложение Б. Код создания основного игрового виджета

```

/**
 * @brief Создать основной игровой виджет
 * @param name Имя игрока
 * @param mode Режим игры
 * @return Виджет для игры. Его класс зависит от режима игры
 */
PlayWidget * GameWindow::createPlayWidget(const QString &name, const QString &mode)
{
    if (mode == tr("Hard")) {
        return new PlayWidgetHard(name, mode, this);
    } else {
        return new PlayWidget(name, mode, this);
    }
}

```

}