

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

Направление подготовки бакалавриата

09.03.04 — Программная инженерия

Отчет по практике

РАЗРАБОТКА ПРИЛОЖЕНИЯ
«LANGUAGE CARDS»

Выполнила:

Хуснутдинова Айгуль Рустемовна студентка 1 курса груп-
пы 22107

А. Р. Хуснутдинова _____
подпись

Петрозаводск — 2020

Содержание

Введение	3
1 Требования к приложению	4
2 Проектирование приложения	5
3 Реализация приложения	7
Заключение	9

Введение

Для изучения иностранных языков зачастую применяется метод изучения новых слов при помощи словарных карточек. Language Cards - приложение, позволяющее пользователю создавать свои словарные карточки, объединять их в колоды, а также иметь возможность проверять своё знание слов при помощи практических заданий.

Цель: проектирование и разработка приложения для изучения иностранных языков по словарным карточкам

Задачи:

1. составить требования к приложению
2. спроектировать дизайн приложения
3. спроектировать архитектуру приложения
4. реализовать приложение

1 Требования к приложению

Требования к приложению с точки зрения пользователя:

1. Понятный и доступный интерфейс
2. Получение статистики после выполненной практики в изучении карточек
3. Преимущественная практика тех карточек, с которыми пользователь справляется хуже всего

Основные функции приложения:

1. Сохранение добавленных карточек и колод после выхода из приложения
2. Добавление и удаление колод
3. Добавление и удаление словарных карточек
4. Поиск слова среди карточек колоды
5. Смена текущей колоды
6. Отображение выбранной пользователем карточки
7. Ведение статистики для каждого слова по количеству правильных ответов
8. Возможность сброса статистики о выученных словах
9. Запуск упражнения, подбирающего слова на основе предыдущих ответов пользователя
10. Запрос подтверждения важных действий (удаление колоды или карточки, сброс статистики, сохранение при выходе из приложения)

2 Проектирование приложения

1. main.cpp - модуль для запуска виджета

- `int main(int argc, char *argv[])` - запуск главного окна

2. mainwindow.cpp - модуль для действий с колодами и карточками

- `load_cards()` - загрузка текущих колод и карточек из файла
- `on_change_deck_clicked()` - смена текущей колоды
- `updateDeck()` - отображение карточек и имени текущей колоды при её смене
- `on_search_line_textChanged(const QString &arg1)` - работа с поисковой строкой
- `on_cards_list_currentTextChanged(const QString ¤tText)` - отображение содержимого выбранной карточки
- `on_change_deck_clicked()` - смена текущей колоды
- `on_add_deck_clicked()` - добавление новой колоды
- `on_remove_deck_clicked()` - удаление текущей колоды
- `on_add_card_clicked()` - добавление новой карточки
- `on_remove_card_clicked()` - удаление новой карточки
- `on_actionQuit_triggered()` - сохранение изменений и выход из приложения
- `on_start_practise_clicked()` - вызов функции запуска диалогового окна при нажатии пользователем кнопки старта практики

3. practise.cpp - модуль запуска упражнения и хранения статистики

- `load_stats()` - загрузка текущей статистики из файла
- `practise()` - вызов диалогового окна для выбора действий
- `play()` - запуск упражнения
- `reset()` - обнуление статистики для данной колоды
- `removeDeckStats(QString cd)` - удаление статистики удалённой колоды
- `addDeckStats()` - добавление статистики для добавленной колоды
- `removeWordStats(QString cd, QString cw)` - удаление статистики для удалённого слова

- `addWordStats(QString cd, QString cw)` - добавление статистики для добавленного слова
- `saveStats()` - сохранение статистики в файл

3 Реализация приложения

Для работы приложения используется набор глобальных переменных, значения которых обновляются при изменении данных пользователем.

При запуске приложения в структуры данных загружаются данные из файла с информацией о колодах и карточках, который построен следующим образом: сначала записано название колоды, начинающееся со знака пробела, затем - слово и перевод, каждый в отдельной строке. При выходе из приложения и сохранении изменений данные из переменных заново загружаются в файл в том же формате.

Для хранения статистики используется та же схема, что и для хранения карточек, однако вместо перевода для каждого слова записываются два числовых значения - количество правильных ответов для данной карточки и количество ответов всего. Упражнение представляет собой набор из 10-ти слов, перевод которых пользователю предстоит ввести. На основе данных статистики происходит сортировка всех карточек: в первую очередь появляются карточки с наихудшими ответами, далее - карточки, ответов на которые ещё не было, и затем - карточки по увеличению доли правильных ответов для них.

Для разработки приложения был использован язык C++, а также следующие библиотеки Qt:

- <QMainWindow>
- <QDebug>
- <QString>
- <QTextStream>
- <QInputDialog>
- <QDir>
- <QMessageBox>
- <QListIterator>
- <QPushButton>
- <QObject>
- <QDialogButtonBox>

- <QApplication>

Оценка сложности разработки:

- Число модулей: 3
- Число функций: 22
- Число исходных файлов: 9

Разработка интерфейса:

Для разработки интерфейса использовались методы Qt.

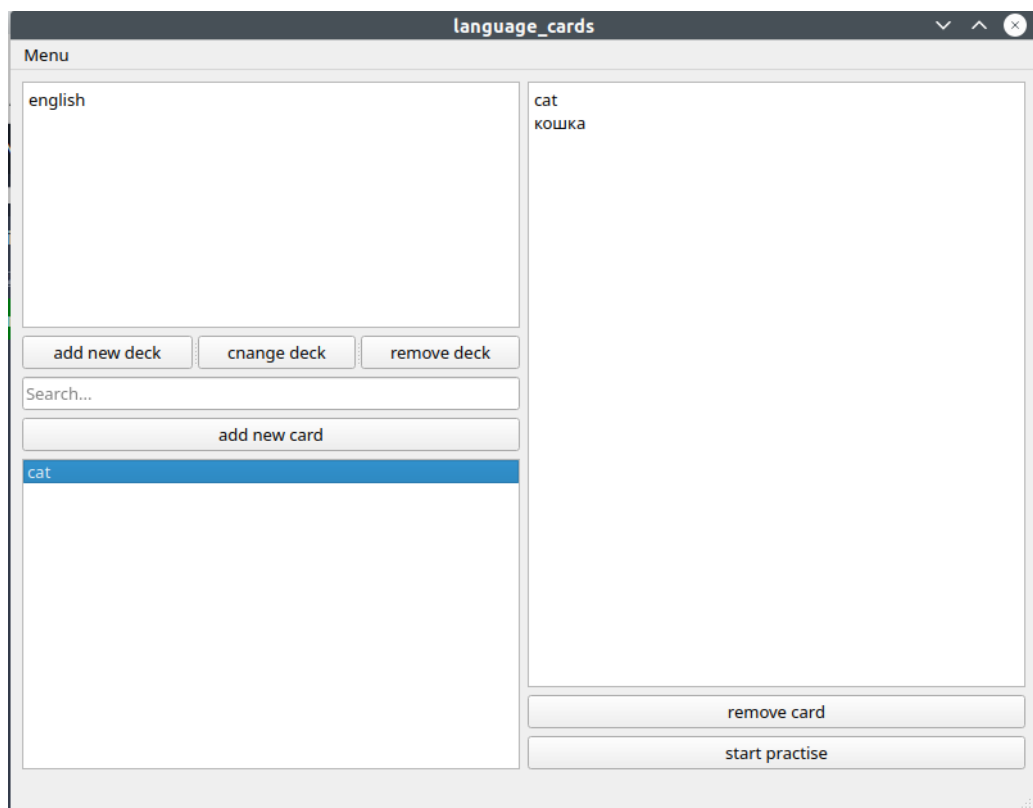


Рис. 1: Главное окно интерфейса

Заключение

Таким образом, в ходе создания приложения были реализованы его основные функции: возможность добавления и удаления колод и словарных карточек, их просмотр. Было реализовано упражнение, проверяющее знания пользователя. Были выполнены основные требования к приложению:

- Интерфейс приложения достаточно прост и понятен пользователю, выдаёт сообщения при некорректном поведении, а также просит подтверждения при выборе пользователем важных действий
- После каждого практического упражнения пользователь получает сообщение о том, сколько правильных ответов он дал, а также сколько слов в упражнении было всего
- Благодаря сортировке всех карточек по доле правильных ответов на них пользователь получает в качестве упражнения прежде всего те слова, запомнить которые получается хуже всего