



МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ  
М. В. ЛОМОНОСОВА

Факультет Вычислительной Математики и Кибернетики  
Кафедра Алгоритмических Языков

Отчёт о выполнении задания практикума  
**«Приложение Anki»**

*Студент 325 группы*  
Кудисов А.А.

Москва, 2022

# 1 Постановка задачи

*Anki* — это приложение, основной целью которого является помочь пользователю выучить новые иностранные слова (или целый фразы) с помощью интервальных повторений:

- Интервальные повторения - техника запоминания, заключающаяся в повторении материала через все возрастающие периоды. В случае трудностей с запоминанием, интервалы напротив равны минимальному значению. Таким образом, наиболее трудный материал повторяется наиболее часто.

## 1.1 Базовые требования

- Механизм интервальных повторений
- Простой интерфейс для создания и управления как самими карточками, так и целыми наборами
- Сохранение пользовательской информации

# 2 Модули проекта

Проект состоит из следующих частей:

- **Main.hs** - основной модуль игры, запуск приложения
- **Config.hs** - константы для настроек игры
- **Utils.hs** - модуль с различными дополнительными функциями для удобства
- **Button.hs** - модуль, содержащий спецификацию объекта Button
- **Card.hs** - модуль, содержащий спецификацию объектов Card и CardCollection
- **State.hs** - модуль, содержащий спецификацию объекта State (состояние всей программы)
- **Graphic.hs** - модуль, отвечающий за отрисовку всего приложения
- **HandleEvent.hs** - модуль, отвечающий за обработку событий
- **Display.hs** - модуль, инициализирующий приложение

Рассмотрим каждый модуль подробнее.

## 2.1 Config.hs

Здесь описаны следующие константные значения:

- `buttonWidth` - ширина кнопок
- `buttonHeight` - высота кнопок
- `collectionWidth` - ширина наборов
- `collectionHeight` - высота наборов
- `pad` - размер отступа между различными элементами интерфейса
- `collectionX` - положение по оси X центра любого набора
- `collectionY` - начальное положение по оси Y центра любого набора
- `menuWidth` - ширина правого меню
- `menuHeight` - высота правого меню
- `menuX` - положение по оси X центра правого меню
- `menuY` - положение по оси Y центра правого меню
- `subMenuWidth` - ширина правого сабменю
- `subMenuHeight` - высота правого сабменю
- `subMenuX` - положение по оси X центра правого сабменю
- `subMenuY` - положение по оси Y центра правого сабменю

## 2.2 Card.hs

Этот модуль содержит следующие типы:

- `Card` - базовая структура "Карточка" <Слово> <Его перевод> <Время последнего повторения> <Рейтинг>
- `CardCollection` - структура "Набор карточек" <Название набора> <Путь до> <Соответствующий объект "кнопка">

И описаны следующие функции:

- `cardToString` - представляет карточку в виде строки
- `stringToCard` - инициализирует карточку из строки
- `isNeedToLearn` - проверяет, нужно ли повторить данную карточку
- `writeCollectionToFile` - сохраняет информацию о наборе в файл
- `writeNewCollectionToFile` - сохраняет информацию о новом наборе в файл и инициализирует создание объекта в системе

- `loadCollection` - загружает набор из файла
- `reloadCollection` - перезагружает набор из файла
- `loadAllCollections` - загружает все известные наборы
- `addCardToCollection` - добавляет карточку в набор
- `updateCollectionPositions` - обновляет позиции наборов (на экране)
- `findNextCardToLearn` - ищет в наборе следующую карточку для обучения

## 2.3 Button.hs

В модуле **Button.hs** объявлены следующие типы:

- `Button` - базовая структура "Кнопка" <Надпись> <Положение центра по X> <Положение центра по Y> <Размер>
- `Size` - структура "Размер" <Радиус> | <Ширина> <Высота>

Также описана следующая функция:

`isButton` - проверяет, было ли нажатие на кнопку

И объявлены следующие кнопки:

- `addButton` - запускает меню создания нового набор
- `upButton` - возвращает список наборов в исходное состояние
- `addCardButton` - запускает меню создания новой карточки
- `saveButton` - сохраняет новую карточку
- `cancelButton` - отменяет действие в рамках набора
- `deleteButton` - удаляет набор
- `stopButton` - останавливает обучение
- `learnButton` - запускает процесс обучения
- `changeNameButton` - меняет имя набора
- `saveChangeNameButton` - сохраняет измененное имя набора
- `cancelChangeNameButton` - отменяет изменение имени набора
- `cancelCardButton` - отменяет действие в рамках карточки
- `addCardWordButton` - переключает ввод на ввод слова
- `addCardTranslationButton` - переключает ввод на ввод перевода
- `showTranslationButton` - показывает перевод

- `easyButton` - увеличивает рейтинг карточки. Слово считается успешно изученным
- `normalButton` - не изменяет рейтинг карточки. Слово считается успешно изученным
- `hardButton` - сбрасывает рейтинг карточки. Слово считается проваленным
- `finishButton` - заканчивает обучение
- `deleteCardButton` - удаляет карточку
- `editCardButton` - запускает редактирование карточки
- `collectionButton` - кнопка набора; выделяет его, открывая всю информацию

## 2.4 State.hs

В этом модуле объявлены следующие типы:

- `Mode` - основное состояние приложения `<Ничего> | <Создается новый набор> | <Выделен набор>`
- `SubMode` - дополнительное состояние приложения `<Ничего> | <Добавляется новая карточка> | <Процесс обучения> | <Редактируется набор> | <Редактируется карточка>`
- `State` - базовая структура, описывающая состояние приложения; `<Путь до наборов> <Наборы> <Стереть> <Основное состояние> <Дополнительное состояние> <Имя новой коллекции> <Слово новой карточки> <Перевод слова новой карточки> <Выбранный набор> <Ширина окна> <Высота окна> <Смещение по Y наборов>`

И описаны следующие функции:

- `stateUpdate` - обновляет состояние приложения
- `getState` - инициализирует начальное состояние
- `resetState` - сбрасывает состояние
- `resetSubState` - сбрасывает дополнительное состояние
- `getSelectedIndex` - получает индекс выбранного набора

## 2.5 Utils.hs

В модуле `Utils.hs` описаны следующие вспомогательные функции:

- `getTime` - получает текущее время
- `dropByIndex` - удаляет элемент из списка по заданному индексу
- `changeByIndex` - изменяет элемент в списке по заданному индексу
- `deleteLast` - удаляет последний элемент из списка
- `wordsWhen` - делит строку по заданному разделителю

## 2.6 Display.hs

В модуле **Display.hs** описаны следующие функции:

- `run` - инициализирует состояние и запускает приложение
- `displayState` - запускает отрисовку приложения

## 2.7 Main.hs

В модуле **Main.hs** описана всего одна функция `main`, запускающая `run` из **Display.hs**

## 2.8 Graphic.hs

Этот модуль состоит из следующих функций:

- `drawState` - основная функция, рисующая всё приложение
- `drawLeftMenu` - рисует левое меню
- `drawRightMenu` - рисует правое меню
- `drawRightSubMenu` - рисует правое сабменю
- `drawLearnContext` - отрисовывает процесс обучения
- `drawCardSettings` - отрисовывает меню карточки
- `drawCollectionList` - отрисовывает список наборов
- `drawCollection` - рисует отдельный набор
- `drawButton` - рисует кнопку
- `drawButtonText` - рисует текст кнопки
- `draw...Button` - ряд функций, отрисовывающих соответствующие им кнопки

## 2.9 HandleEvent.hs

Данный модуль содержит следующие функции:

- `is...Button` - ряд функций, проверяющих, что соответствующая кнопка нажата
- `isCollection` - проверяет, было ли нажатие на один из наборов
- `handleEvent` - основная функция, следящая за событиями
- `saveNewCollection` - сохраняет новую коллекцию
- `deleteCollection` - удаляет набор
- `saveCard` - сохраняет карточку
- `saveNewCard` - сохраняет новую карточку

- `cancelCard` - сбрасывает дополнительное состояние
- `editCard` - переводит дополнительное состояние в редактирование карточки
- `saveChangeCard` - сохраняет изменения карточки
- `activateLearnSubMode` - начинает процесс обучения
- `nextOrEndLearn` - ищет следующая карточку для обучения или завершает его
- `nextLearn` - ищет следующую карточку для обучения
- `changeRating` - изменяет рейтинг карточки
- `changeCollectionName` - изменяет имя набора
- `deleteCard` - удаляет карточку
- `changeEnterMode` - переключает между вводом слова и вводом перевода этого слова

### 3 Используемые библиотеки

При реализации использовались следующие библиотеки:

- `gloss` - графический интерфейс и обработка внешних событий
- `directory` - для работы с файловой системой
- `time` - для работы со временем

### 4 Сценарии работы с приложением

При запуске программы пользователь видит разделенный на две половины экран. Слева отображаются уже существующие колоды/наборы карточек, а также кнопка создания новой колоды. Дополнительно реализован механизм пролистывания колод ввиду их возможно большого числа.

При нажатии на любую из колод с правой стороны отобразится информация о ней, а также возможные дальнейшие действия: удалить колоду, изменить имя, начать обучение, добавить карточку.

Пользователь может начать обучение, в таком случае из колоды ему будут по очереди показаны все слова, которые необходимо повторить на данный момент. Если обучение не вызывает никаких проблем, то пользователю следует нажать на кнопку "Easy", таким образом рейтинг соответствующей карточки будет повышен и в следующий раз она будет показана через больший интервал времени. В случае возникновения проблем, пользователь может нажать "Normal" или "Hard", в результате рейтинг не изменится (интервал остенется тем же) или же станет равен -1 (интервал станет равен 0 дней) соответственно. Как только в колоде не останется не изученных на данный момент карточек, обучение закончится.