Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра информатики

Курсовой проект по дисциплине:

«Программирование»

Пояснительная записка к курсовой работе

Тема работы:

«Приложение-тренажёр для эффективного запоминания информации»

Исполнитель

студентка гр. 653502 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Дейкун Е. В.

(подпись дата)

Руководитель

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Козуб В.Н.

(подпись дата)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(оценка)

Минск

2017 год

Содержание

[Введение 3](#_Toc482035699)

[1. Постановка задачи 4](#_Toc482035700)

[2. Анализ аналогичных приложений 5](#_Toc482035701)

[3. Выбор средств разработки программы 8](#_Toc482035702)

[3.1 Описание языка C # 8](#_Toc482035703)

[3.2 Базы данных MySQL 9](#_Toc482035704)

[4 Алгоритм работы программы 11](#_Toc482035705)

[5. Руководтсво пользователя 13](#_Toc482035710)

[Выводы 17](#_Toc482035711)

[Список использованных источников 18](#_Toc482035712)

## Введение

В современном мире каждый образованный человек должен знать не менее двух языков. На их изучение нужно много времени, а заниматься приходится каждый день. Одним из основных аспектов в изучении иностранного языка является пополнение своего словарного запаса. Сейчас для эффективного запоминания и повторения слов широко используется метод с помощью карточек по системе Лейтнера. Эта система — простое применение принципа интервальных повторений, где карточки повторяются через увеличивающиеся интервалы. Флеш-карточки - это мощный инструмент для запоминания, потому что самым неэффективным методом для запоминания является перечитывание материала. Это создает иллюзию знакомства с информацией, в то же время воспроизведение этой информации окажется невозможным. Для того чтобы информация откладывалась в памяти можно использовать флеш-карточки. Флеш-карточки могут быть бумажными, аналоговыми и электронными.

Делать такие карточки в бумажном варианте крайне неудобно. Решением данной проблемы становится разработка системы, которая будет реализовывать этот метод через электронные карточки. Положительные стороны такой организации карточек заключаются в следующем:

1. Возможность создавать более наглядные карточки и редактировать их позже
2. Пользователю не приходится вручную перекладывать карточки из контейнера в контейнер, программа делает это автоматически
3. Есть возможность просматривать карточки в игровой форме, что делает обучение более интересным
4. Возможность синхронизировать их на различных устройствах
5. Возможность использования изображений, видео, аудио при создании карточек

Изучение иностранных языков – только один пример применения карточек. С их помощью можно запоминать любую информацию, которую можно представить в таком виде. Такая система широко используется среди студентов медицинских университетов, так как им нужно запоминать огромное количество информации и иметь возможность без проблем воспроизвести ее через несколько лет.

## 1. Постановка задачи

Система Лейтнера основана на принципе интервального повторения, который заключается в том, чтобы повторять материал именно тогда, когда он немного забылся, потому что именно в этот момент мы перестаем воспроизводить информацию из краткосрочной памяти,  а вместо этого усиливаем нейронные связи с новой информацией.

Все карточки условно разделяются на контейнеры. Только что добавленные помещаются в первый контейнер (всего их 5). Если при прохождении теста пользователь вспомнил информацию, то карточка перемещается в следующий контейнер, если же он ее не вспомнил, то карточка перемещается на контейнер ниже. Иллюстрация такого процесса показана на рисунке 1.



Рисунок 1.

Таким образом, пользователь имеет набор карточек, распределенный по контейнерам. Карточки из первого контейнера следует повторять каждый день, из второго – раз в 2-3 дня, из третьего – раз в неделю, из четвертого – раз в две недели, из пятого – раз в месяц. Суть такого процесса – переместить все карточки в пятый контейнер.

Если использовать электронный вариант, то о перемещении карточек из контейнера в контейнер даже не приходиться задумываться, потому что программа делает это самостоятельно.

Реализация такого способа использования флэш-карточек и стала главной задачей моей курсовой работы.

## 2. Анализ аналогичных приложений

Для разработки качественного продукта важным этапом является анализ подобных приложений и выявление их недостатков. Далее будут проанализированы наиболее популярные на данный момент приложения, которые реализуют метод интервальных повторений.

**Anki**

*Функции*

Основу базы данных Anki , хранящейся в формате SQLite, составляет список записей. Каждая запись является набором полей со словами, определениями, произношениями и т. п., из которых по указанным в базе шаблонам автоматически создаются карточки. Поля могут состоять из HTML со стилями, изображениями, звуками, видео и LaTeX.

Anki может делать так, чтобы разные карточки, сгенерированные из одной записи (например, «картинка → слово» и «слово → картинка»), не появлялись подряд (в версии 1.x время настраиваемо; с версии 2.0.21 они по умолчанию откладываются до завтра, отложенные можно вернуть заранее). Доступна для пользователя настройка и многих других переменных.

Если при оценке ответа допущена ошибка, оценку можно отменить.

Доступно некоторое количество статистики в виде цифр и графиков.

Были специальные функции для изучения японского языка — например, при вводе в поле «вопрос» кандзи в поле «ответ» может автоматически вводиться соответствующая фуригана. С версии 0.9.9.8.2 функции для японского и китайского языков вынесены в плагины.

Anki поддерживает синхронизацию с сервером, регистрация на котором бесплатна. Размещая «колоду» карточек на сервере, её можно синхронизировать с несколькими устройствами или заучивать, используя сайт (AnkiWeb).

*Алгоритм*

Первые версии Anki использовали алгоритм SuperMemo SM5. Однако, по словам авторов, обнаружилось, что алгоритмы SM3 и новее, в которых, в отличие от SM2, ответ на одну карточку влияет на время появления других, похожих на неё, на самом деле имеют значительные недостатки по сравнению с SM2, в котором при ответе устанавливается промежуток только для данной карточки. Это особенно заметно при нерегулярном обращении к программе. Теперь Anki использует алгоритм, основанный на SM2.



Рисунок 2. Anki

**Quizlet**

Quizlet - онлайн сервис, с помощью которого можно создавать флэш-карточки для любой предметной области. Предполагает режим тренировки новых знаний, закрепления и контроля. Тут создается набор карточек, которыми   можно поделиться с другими пользователями. Так, например, учитель может создать коллекцию и поделиться ею с учениками. После создания карточек можно выбрать режим работы, где доступны тренировки по типу: учить, орфограф (слушаем и вписываем фразу в поле), собирать рассыпанные фразы, пройти тест (вписать ответ, выбрать ответ, выбрать "правильно" - "неправильно").

Quizlet также поддерживает добавление мультимедийных файлов.

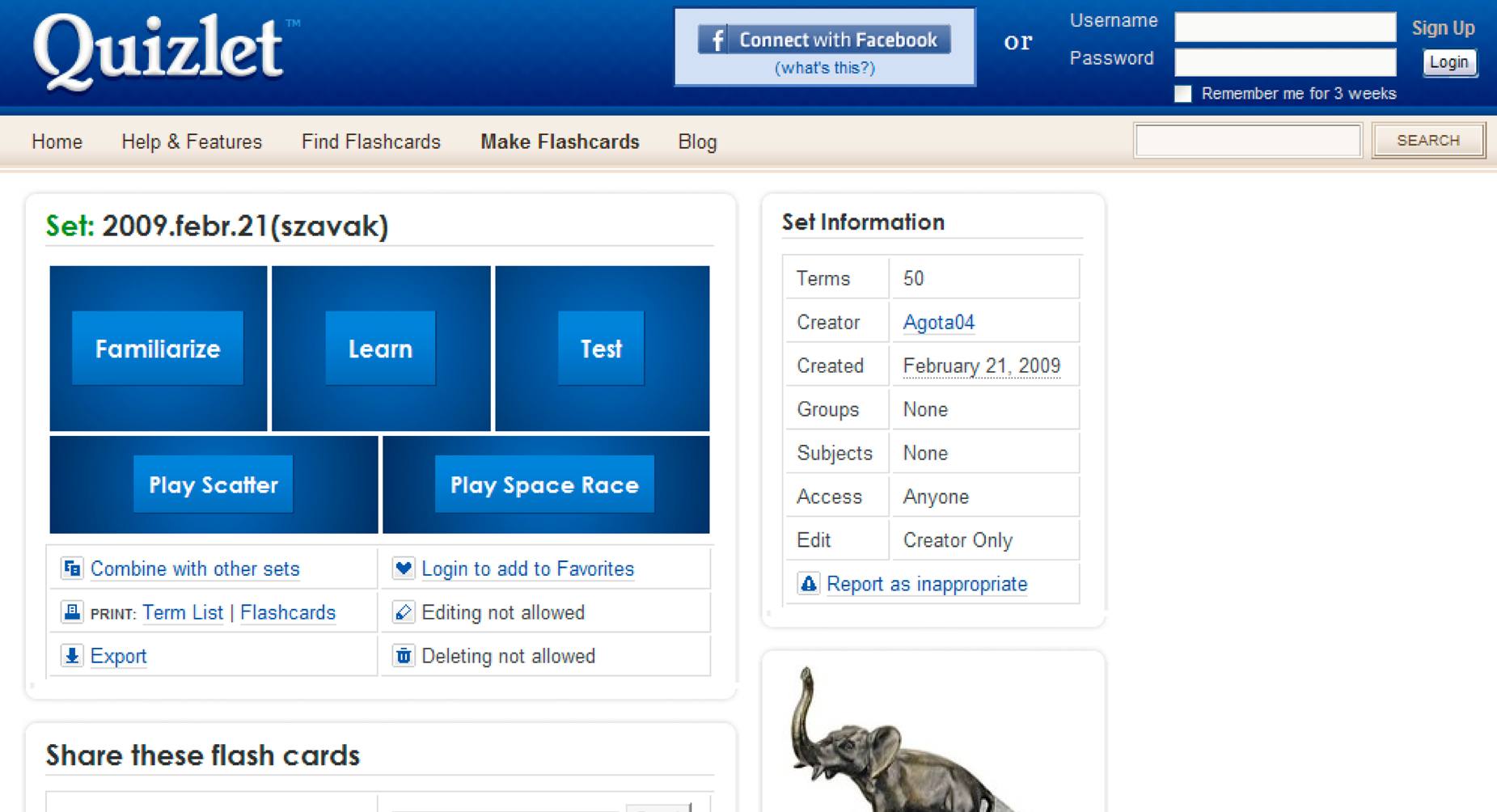


Рисунок 3. Quizlet

*Основные возможности этих приложений*:

* Широкие возможности по настройке вида карточек
* Поддержка мультимедиа
* Большое количество плагинов
* Настройка алгоритма обучения
* Подробная статистика
* Поддержка большинства современных платформ (Linux, Windows, Mac, iPhone, Android)
* Синхронизация данных

*На мой взгляд, минусы данных приложений заключаются в следующем:*

1. Сложный интерфейс, слишком большое количество различных функций, что рассеивает внимание. На приспособление к ним уходит очень много времени.
2. Наличие возможности делиться своими карточками с другими людьми, а так же добавлять их карточки к себе, потому что процесс создания карточек очень специфичен. Он требует детального разбора материала, разбиения его на мелкие пункты, выделение наиболее важной для себя информации. Только после этого создаются карточки. Если же просто скачать чужие, у вас не будет тех знаний, которые приобрели эти люди в ходе их создания.

## 3. Выбор средств разработки программы

Для разработки данной программы я выбрала интегрированную среду разработки Microsoft Visual Studio 2017. Данная среда позволяет разрабатывать как консольные приложения, так и приложения с графическим интерфейсом, в том числе с поддержкой технологии Windows Forms, а также веб-сайты, веб-приложения, веб-службы как в родном, так и в управляемом кодах для всех платформ, поддерживаемых Windows, Windows Mobile, Windows CE, .NET Framework, Xbox, Windows Phone .NET Compact Framework и Silverlight.

Visual Studio включает в себя редактор исходного кода с поддержкой технологии IntelliSense и возможностью простейшего рефакторинга кода. Встроенный отладчик может работать как отладчик уровня исходного кода, так и отладчик машинного уровня. Остальные встраиваемые инструменты включают в себя редактор форм для упрощения создания графического интерфейса приложения, веб-редактор, дизайнер классов и дизайнер схемы базы данных. Visual Studio позволяет создавать и подключать сторонние дополнения (плагины) для расширения функциональности практически на каждом уровне, включая добавление поддержки систем контроля версий исходного кода (как, например, Subversion и Visual SourceSafe), добавление новых наборов инструментов (например, для редактирования и визуального проектирования кода на предметно-ориентированных языках программирования) или инструментов для прочих аспектов процесса разработки программного обеспечения (например, клиент Team Explorer для работы с Team Foundation Server).

**3.1 Описание языка С#**

**C#** — объектно-ориентированный язык программирования, относится к семье языков с C-подобным синтаксисом, из них его синтаксис наиболее близок к C++ и Java. Язык имеет статическую типизацию, поддерживает полиморфизм, перегрузку операторов (в том числе операторов явного и неявного приведения типа), делегаты, атрибуты, события, свойства, обобщённые типы и методы, итераторы, анонимные функции с поддержкой замыканий, LINQ, исключения, комментарии в формате XML.

**C#** находит свое применение в **многочисленных сферах программной индустрии**. Он был разработан и по сегодняшний день активно развивается гигантов программной индустрии, компанией Microsoft. Знания языка **C# позволяют разрабатывать практически любые программные продукты,** это могут быть: десктоп, серверные, мобильные приложения, игры, сложные вычислительные программы, можно разрабатывать даже приложения для Android, IOS и Линукс систем.

**Microsoft заботится о своевременном обновлении документации** как самого языка, так и связанных с ним технологий, таких как .Net ( обязательная составляющая практически любого приложения для Windows). .Net – это фреймворк, который содержит все необходимые базовые классы и методы для работы с файлами, базой данных, сетью, операционной системой, содержит механизмы криптографии и сериализации данных. В общем, содержит все базовые и не только инструменты для создания программ на C#. И это большой плюс, потому как понимание основных принципов .Net полезно на при разработке как десктопных, так и серверных и мобильных приложений. А значит, везде где C# и Windows – там и .Net.

Также **C# активно поддерживается различными сообществами**. На одном из самых популярных сайтов о решении вопросов разработки программных продуктов [StackOverflow](http://stackoverflow.com/) уже существует 883,637 вопросов-ответов на различные области, связанные с C#, в то время как про Python создано всего 507,542 вопрос-ответов.

**3.1 База данных MySQL**

Все приложение построено на использовании карточек, которые входят в определенную коллекцию, содержат ключевое слово, описание и изображение. Проблемой стал выбор способа хранения этой информации. Ранее для долговременного хранения информации я работала с файлами: помещала в них некоторое количество строчек, а затем извлекала их для последующей работы. Задача длительного хранения информации очень часто встречается в программировании Web-приложений: подсчёт посетителей в счётчике, хранение сообщений в форуме, удалённое управление содержанием информации на сайте и т.д.

Между тем, профессиональные приёмы работы с файлами очень трудоёмки: необходимо заботится о помещении в них информации, о её сортировке, извлечении, нужно так же заботится о правах доступа к файлам и их размещении. При этом объём кода значительно возрастает, и совершить ошибку в программе очень просто.

Все эти проблемы решает использование базы данных. Базы данных сами заботятся о безопасности информации и её сортировке и позволяют извлекать и размещать информацию при помощи одной строчки. Код с использованием базы данных получается более компактным, и отлаживать его гораздо легче. Кроме того, не нужно забывать и о скорости - выборка информации из базы данных происходит значительно быстрее, чем из файлов.

**MySQL** – это одна из самых популярных и самых распространенных СУБД (система управления базами данных) в интернете. Она не предназначена для работы с большими объемами информации, но ее применение идеально для приложений, как небольших, так и достаточно крупных.

**MySQL** отличатся хорошей скоростью работы, надежностью, гибкостью. Работа с ней, как правило, не вызывает больших трудностей. MySQL распространяется на условиях общей лицензии GNU (GPL, GNU Public License).

* 1. **Алгоритм работы программы**

Определившись с выбором языка и способом хранения информации можно приступить к разработке алгоритма работы программы. Поскольку нам требуется хранить и информацию о пользователе и его коллекции, мы используем две таблицы. При регистрации нового пользователя в таблицу вставляется запись о нем. Таблица пользователей содержит информацию о всех пользователях и она организована, как показано на рисунке:

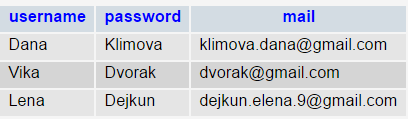


Рисунок 4.

При регистрации также создается таблица с названием пользователя, в которой хранятся его коллекции. Таблица имеет следующую структуру:

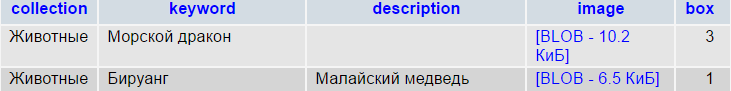


Рисунок 5.

Далее вся работа с карточками и коллекциями основана на использовании таблиц. На вкладке «Профиль» пользователь может просмотреть свою учетную запись и изменить информацию. Она обновляется в базе данных.

Пользователь имеет возможность добавлять карточки в коллекции, просматривать коллекции, изменять карточки. Такие функции реализованы на вкладке «Коллекции».

На вкладке «Тренировки» есть два типа тренировок. Выборка карточек для тренировок составляется с учетом запроса пользователя с помощью команды SELECT. Для первой тренировки выбираются все элементы. Она состоит в заполнении текстового поля ключевым словом, которое соответствует данному изображению. Если пользователь не может вспомнить слово, он может кликнуть по кнопке «Подсказка». Вторая тренировка заключается в следующем: две таблицы заполняются изображениями и ключевыми словами. Задача – соединить все изображения с их ключевыми словами. При совпадении ключевого слова и изображении, соответствующего ему эти два элемента исчезают. Тренировка окончена, когда исчезнут все элементы.

Вкладка «Прогресс» является наиболее значимой, поскольку при прохождении теста именно тут определяется, в какой контейнер отправится карточка. Если пользователь вспомнил ее, то ее уровень поднимается на единицу, а если не вспомнил – опускается на единицу. Все изменения записываются в таблицу. Также тут можно просмотреть информацию о количестве карточек в каждом контейнере.

* 1. **Руководство пользователя**

Для начала пользования приложением достаточно запустить exe-файл. При запуске программы первым делом загружается форма входа, в которую нужно ввести логин и пароль или зарегистрироваться. После входа в систему появляется главная форма программы. На вкладке «Главная» находится информация о системе Лейтнера и советы по формированию карточек. На вкладке «Профиль» находится информация об учетной записи пользования, которую можно изменить.

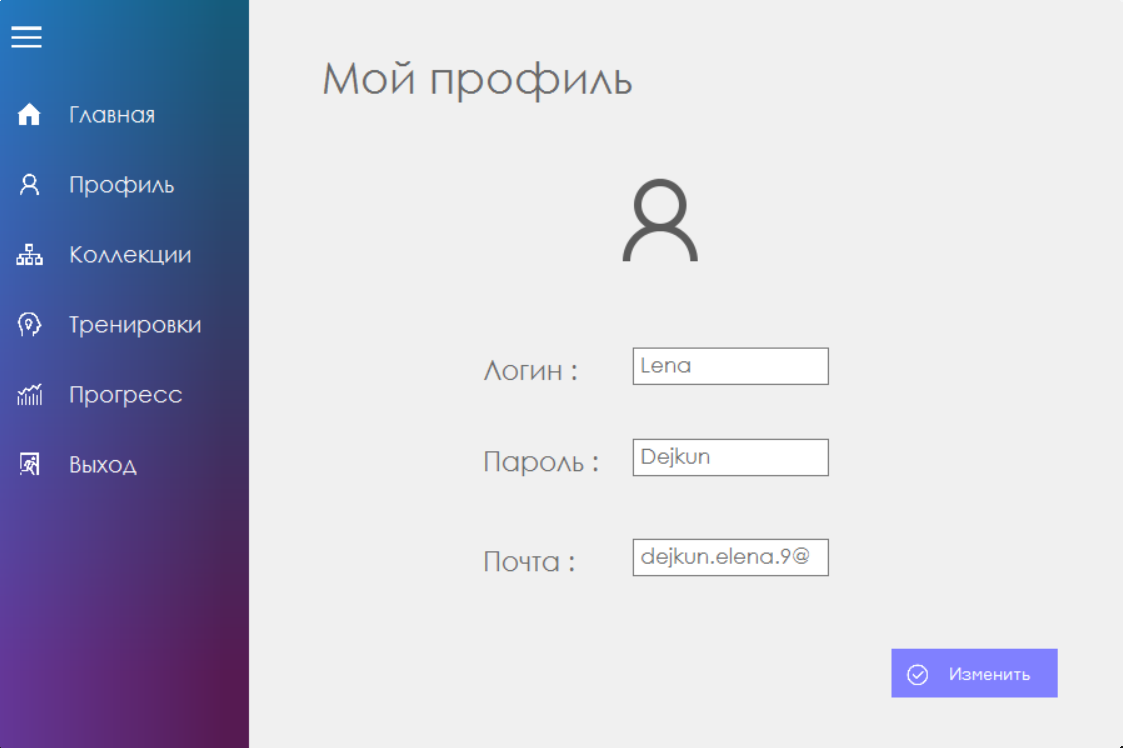


Рисунок 6. Профиль пользователя

На вкладке «Коллекции» находятся коллекции пользователя. Для перехода к перечню карточек коллекции необходимо нажать на нужную коллекцию.

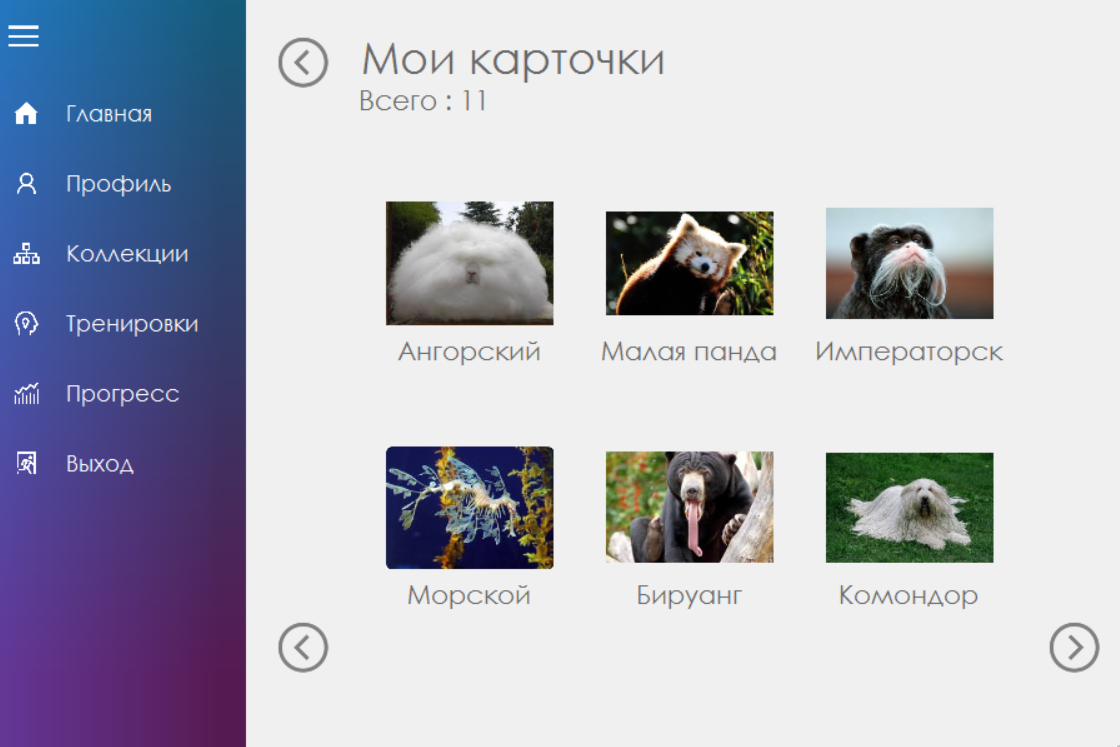


Рисунок 7. Карточки пользователя

Для перехода к карточке и ее изменения необходимо кликнуть на неё. Чтобы добавить новую карточку нужно нажать на кнопку «Добавить». При этом нужно заполнить все поля, кроме поля описания, так как оно является не обязательным полем.

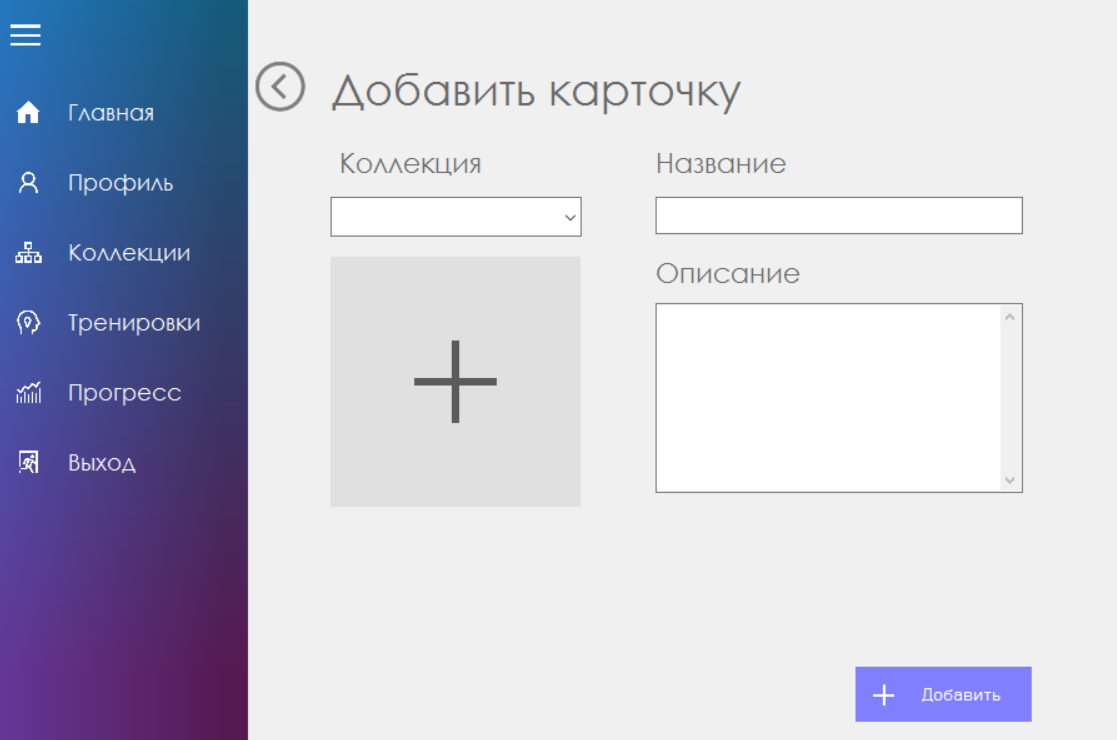


Рисунок 8. Добавление карточки

На вкладке «Тренировки» находится перечень доступных тренировок. Для прохождения каждой из них необходимо выбрать желаемую коллекцию и уровень.

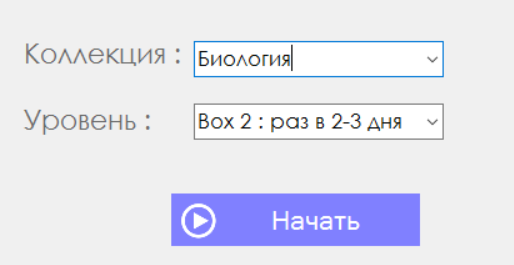


Рисунок 9. Выбор коллекции для тренировки

Первая тренировка. Здесь выводится изображение и необходимо ввести ключевое слово. Для получения подсказок можно кликнуть на изображение.

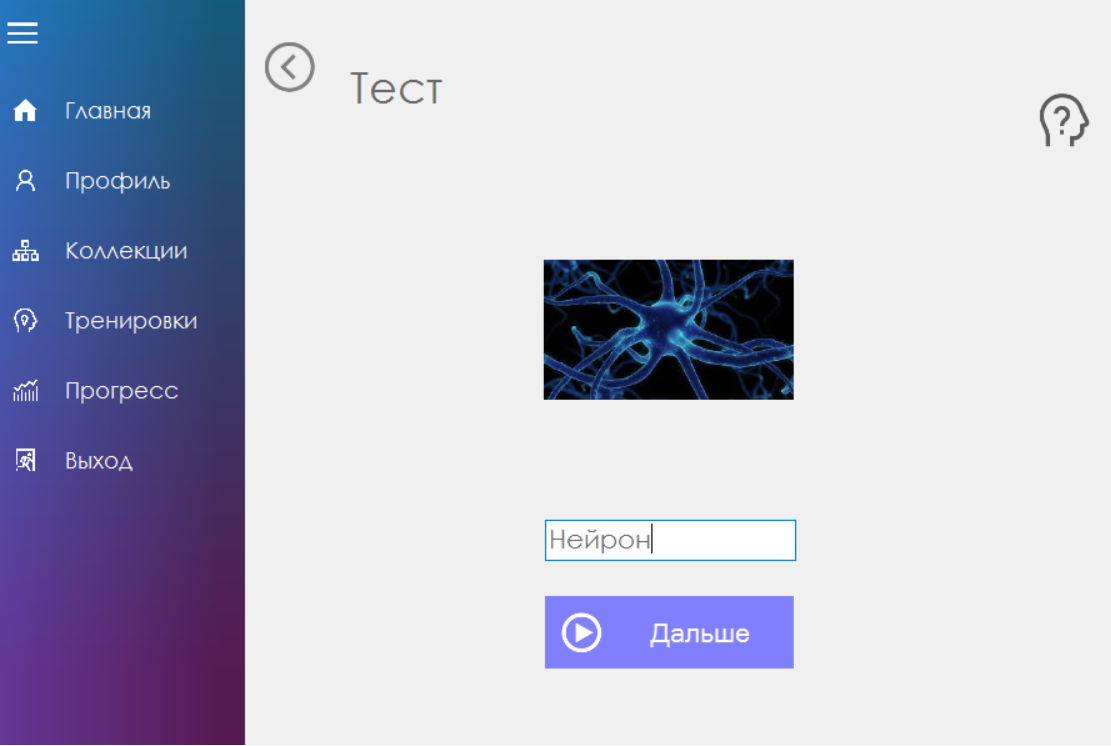


Рисунок 10. Тренировка 1

Вторая тренировка. Здесь нужно попарно кликать на картинку и ключевое слово, если пары будут выбраны неправильно, то они выделятся другим цветом. Если пары выбраны правильно, то они исчезнут.

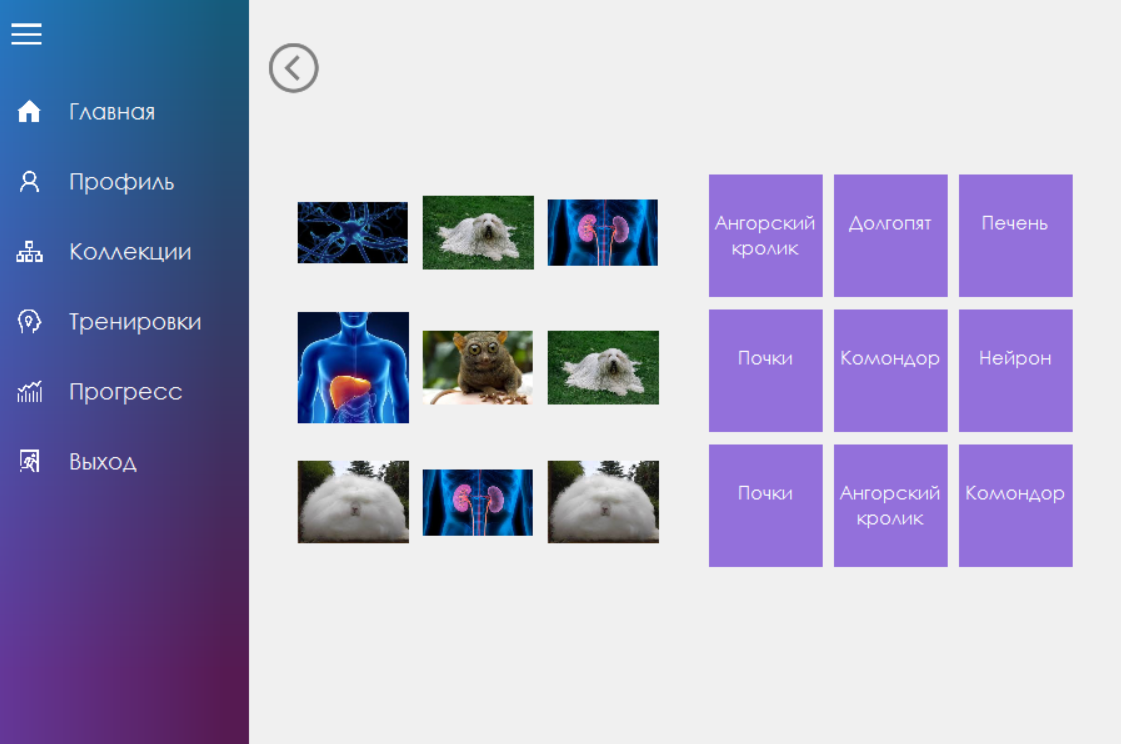


Рисунок 11. Тренировка 2

На вкладке «Прогресс» можно пройти тест, в результате которого будет изменен контейнер, в котором находится данная карточка. Это зависит от того, помните ли вы данное слово или нет. При клике на кнопку «Прогресс» выведется количество карточек в каждом контейнере. Напомню, что цель – собрать все карточки в пятом контейнере. Это значит, что информация усвоилась. При прохождении этого теста также есть возможность проверить себя и получить подсказку, нажав на изображение.

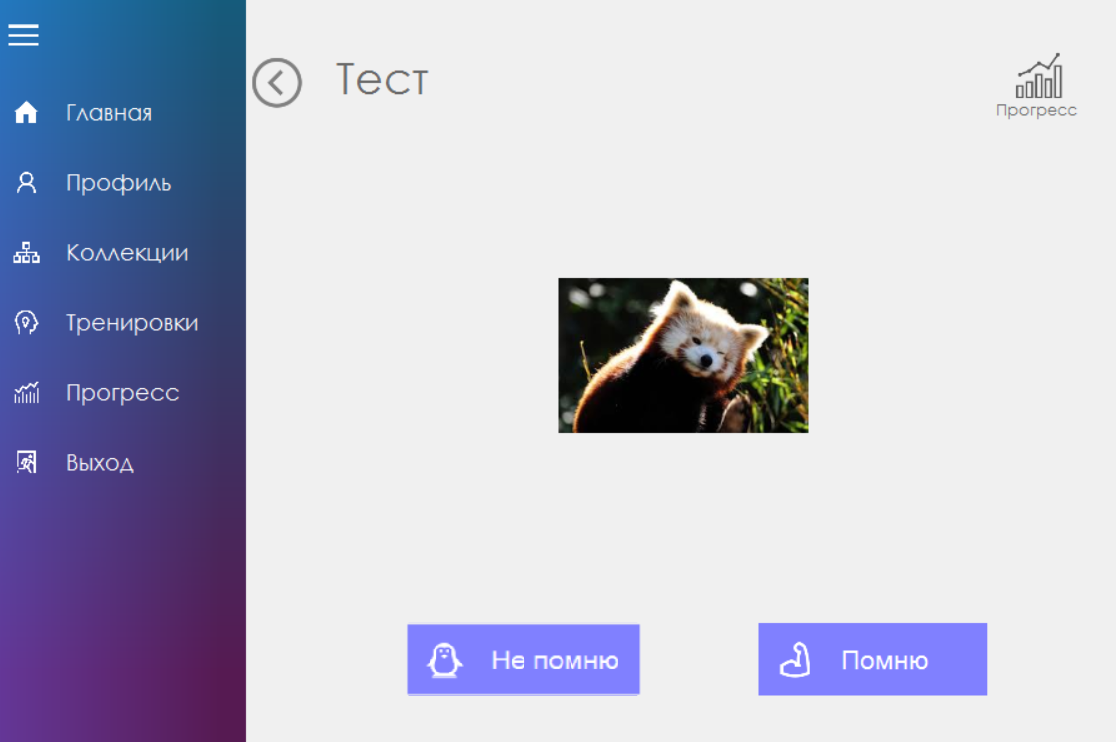


Рисунок 12. Тест

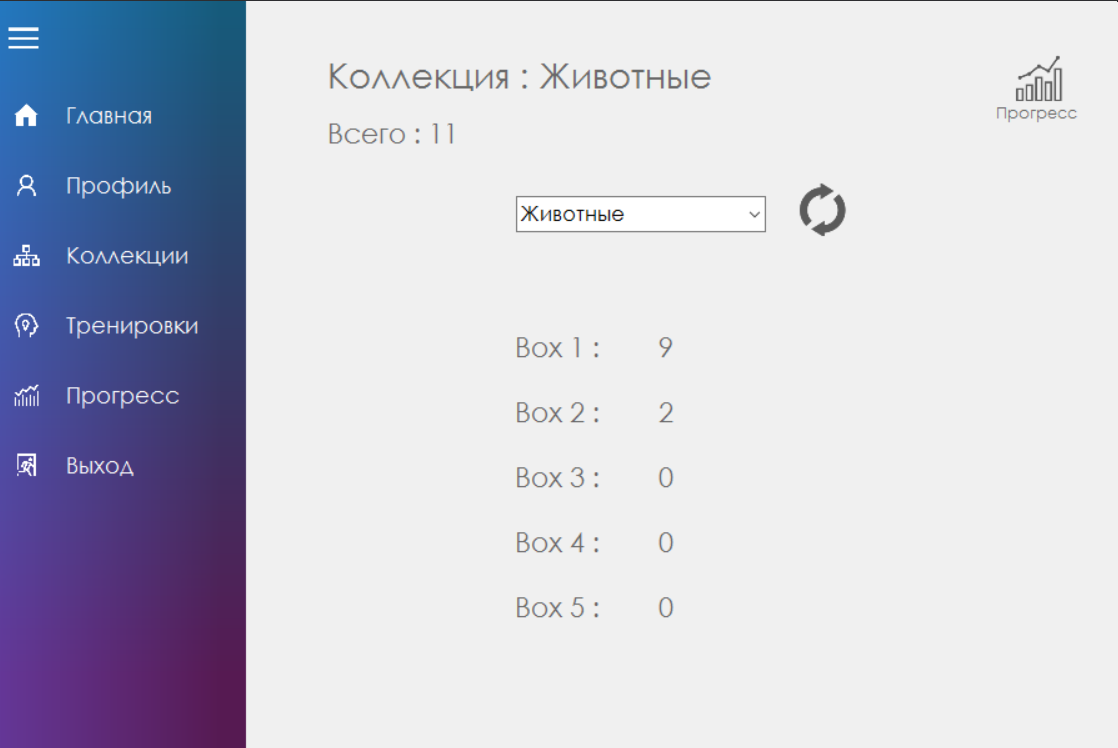


Рисунок 13. Прогресс

## Выводы

В ходе работы над данным курсовым проектом была создана программа для эффективного запоминания информации. Для ее разработки использовался высокоуровневый объектно-ориентированный язык программирования C# и язык управления базами данных Sql.

На данный момент в программе реализована возможность добавления карточек, просмотра коллекций, работа с ними, отслеживается прогресс пользователя.

В данной программе осуществлена главная задача, но она требует дальнейшего развития и доработки. Во-первых, можно расширить возможности создания карточек (путем добавления аудио, видео). Во-вторых, стоит рассмотреть вариант создания интеллект-карт на основе имеющихся флеш-карточек.

## Список использованных источников

1. Система Лейтнера [https://en.wikipedia.org/wiki/Leitner\_system](https://clck.yandex.ru/redir/LvUXD5J6I4o?data=UVZ5S3FTUHlHUXdOdmRvb3dORVBqYUtQUDBhWG5PeHhqT1FyREtIdmRKR0ptZTJNdXlmQnRUemFZWC00YXJ6TlBVOE9DbnpnV0k5MnJvcW5mbjVFV3dRNXBYQkxkTVJwWkR3M00ycWxRRzRmSVREMUVuYWNNV3h4ZklLXzcyVWhwU2RpU19pNEVpVQ&b64e=2&sign=ce8fd4ab6a0c3d0b36bdd8f2122060fa&keyno=1)
2. [https://msdn.microsoft.com/ru-ru/library/](https://clck.yandex.ru/redir/LvUXD5J6I4o?data=UVZ5S3FTUHlHUXdOdmRvb3dORVBqVVJDQVhWRUluRlV3a2ZXMTVLM210WEZTSm1wS0FDR1NLRHo5Q0dPVlZRQ3F4NkM4b05SamotMURrQk9kSEw4VW50ZlZHR0xNMXBBeFdUOGQwV2FRWWFuUGdhY0ZRUlAtekFlcVNvOHdiS0VCZlBYZzVCX3o5aw&b64e=2&sign=14455d2f4dd4a50558b726ea5171aa1b&keyno=1) [Электронный ресурс]
3. [https://docs.microsoft.com/ru-ru/dotnet/](https://clck.yandex.ru/redir/LvUXD5J6I4o?data=UVZ5S3FTUHlHUXdOdmRvb3dORVBqY1FadmZqS29vb19pWGxsQ1QyVS11TlR3c1lKY2p2cDVkMU5KX2FnakVPa1NQN29JaEpBaHQ0U3FlYjFkWFZIMWpPQ2JQcnZrRjItX0tJdzlBWHNyLWMtNE1kLXBjbWRfam9GNUZrNVMtMDRfanNpRzZqZnRhNA&b64e=2&sign=c20bf60b78530bebfb07460091fec8e1&keyno=1) [Электронный ресурс]
4. <https://www.phpmyadmin.net/> [Электронный ресурс]
5. <https://www.mysql.com/> [Электронный ресурс]