Министерство образования Новосибирской области ГБПОУ НСО «Новосибирский авиационный технический колледж имени Б.С. Галущака»

**Разработка мобильного приложения для онлайн бронирования спортивного инвентаря SNOWBALL**

Пояснительная записка к курсовому проекту

ПМ.01Разработка модулей программного обеспечения для компьютерных систем

НАТКиГ.210500.43.000ПЗ

Выполнил: Супрунов А.А.

2021

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc70519047)

[1 ВЫБОР ТЕХНОЛОГИИ, ЯЗЫКА И СРЕДЫ ПРОГРАММИРОВАНИЯ 4](#_Toc70519048)

[2 Проектирование интерфейса приложения 5](#_Toc70519049)

[2.1 Анализ и уточнение требований к программному продукту](#_Toc70519050) 6

[2.2 Проектирование UI и UX дизайна приложения](#_Toc70519051) 7

## 2.3 Разработка форм ввода-вывода информации……………………………...19

[3 Выбор методов и разработка основных алгоритмов решения задачи](#_Toc70519052) 20

[4 Разработка мобильного приложения](#_Toc70519053)

[4.1 Описание используемых процедур и библиотечных функций](#_Toc70519054) 21

[4.2 Спецификация программы](#_Toc70519055) 23

[ЗАКЛЮЧЕНИЕ](#_Toc70519056) 24

[СПИСОК ИСТОЧНИКОВ](#_Toc70519057) 25

[Приложение А](#_Toc70519058) 26

ВВЕДЕНИЕ

В настоящее время бронирование снаряжения у прокатных сервисов осуществляется непосредственным приходом клиента в прокатный сервис, что является не автоматизированным учетом необходимой информации. На данный момент в сфере прокатных сервисов наблюдается недостаточное количество онлайн-сервисов для оформления заказов и просмотра доступной продукции, что является существенным недостатком. Таким образом, актуальность курсового проекта обуславливается периодической потребностью населения в быстром бронировании спортивного снаряжения.

Программный продукт «Бронирования спортивного снаряжения SnowBall» предназначен для применения в сфере услуг с целью онлайн-бранирования спортивного инвентаря.

Разрабатываемый программный продукт позволяет облегчить и ускорить процесс выбора инвентаря для клиентов с помощью доступной системы поиска. Мобильное приложение обеспечивает персонализацию и удобный функционал.

Целью курсового проекта является разработка мобильного приложения для разных групп пользователей, которое включает в себя доступную систему поиска товаров, позволяющую ускорить процесс нахождения и выбора требуемого снаряжения.

Для достижения поставленной цели определены следующие задачи:

* изучение и анализ предметной области;
* проектирование авторского дизайна разрабатываемого приложения;
* разработка клиентской и серверной части базы данных;
* создание структуры гипертекстового документа на основе XML-разметки в соответствии с разработанными макетами;

Предмет исследования курсового проекта — мобильное приложение проката снаряжения.

# ВЫБОР ТЕХНОЛОГИИ, ЯЗЫКА И СРЕДЫ ПРОГРАММИРОВАНИЯ

Во время создания программы выбор языков программирования обуславливался такими критериями, как: скорость создания программы и её тестирования, возможность переноса на другие платформы, возможность быстрого внесения изменений, быстрота выполнения конечного продукта. Покрытием всего востребованного функционала и предоставлением возможности механизмов анализа кода обуславливался выбор среды разработки для программного продукта.

Веб-приложения реализовывалось в среде разработки –Android Studio, являющейся наиболее функционально насыщенной IDE-средой, в которой можно разрабатывать приложения Java и Kotlin. Данный продукт поставляется с конструкторами графических пользовательских интерфейсов, возможностями работы с базами данных, поддержкой фрагментов кода, возможностями для просмотра всего проекта в целом и просмотром свойств объектов.

Android Studio — это современная среда программирования, созданная для тестирования и разработки приложений для ПК под ОС Android. При разработке оболочки были внедрены необходимые инструменты для комфортного, качественного и быстрого создания новых и отладки существующих приложений. Оболочка Android Studio содержит Android SDK, все версии операционных систем Android, эмуляторы для запуска рабочих приложений, а также различные компоненты для тестирования и отладки программных продуктов.

Основным языком программирования был выбран Java, потому что, для любого разработчика мобильных приложений на Android, первым и наиболее предпочтительным языком программирования пока остаётся именно он. Java поддерживается компанией Google и большинство приложений в Google Play построены именно на нем.

Для хранения информации о магазинах была выбрана облачная база данных Firebase, которая позволяет пользователям хранить и получать сохранённую информацию, а также имеет удобные средства и методы взаимодействия с ней. [Firebase](http://firebase.google.com/) хранит текстовые данные в [JSON](https://ru.wikipedia.org/wiki/JSON) формате и предоставляет удобные методы для чтения, обновления и извлечения данных. Также, Firebase может помочь с регистрацией и авторизацией пользователей, хранением сессий (авторизованные пользователи), медиафайлов к которым с лёгкостью предоставляет доступ благодаря Cloud Storage.

# Проектирование интерфейса приложения

## Анализ и уточнение требований к программному продукту

Анализируя курсовой проект выявлены общие требования к разрабатываемому программному продукту, а именно:

Требования к исходным кодам и языкам программирования:

* исходные коды программы должны быть написаны на языке Java. Тип мобильного приложения: мобильного приложение для операционной системы Android, API 21+, Android 5.0+ Lollipop;
* база данных должна быть размещена в Firebase Realtime Database;
* Адаптивное отображение верстки экранов приложения на всех разрешениях (диагоналях);
* наличие двух группы пользователей: администратор пользователи, отвечающие за администрирование данных и управление соответствующей функциональностью мобильного приложения;
* системные программные средства, используемые программой, должны быть представлены операционной системой Android 5.0+ Lollipop;
* Для работы мобильного приложения на смартфоне должен быть активно интернет-соединение.

## Проектирование UI и UX дизайна приложения

На основе технического задания спроектирован макет UI дизайна мобильного приложения в Figma. Это кроссплатформенный онлайн-сервис для дизайнеров интерфейсов и веб-разработчиков. Разработка интерфейсов происходит в онлайн-приложении.

В проектировании использовались техники UX дизайна с целью обеспечения удобного и понятного пользовательского интерфейса. В макете применена следующая цветовая гамма на рисунке 1:

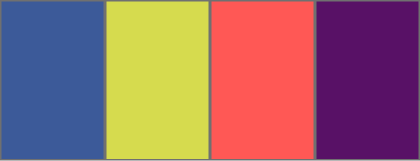


Рисунок 1 – Цветовая гамма макета мобильного приложения

Логотип приложения был разработан с применением той же цветовой гаммы, что и в макете мобильного приложения. Логотип мобильного приложения «SnowBall» показан на рисунке 2.

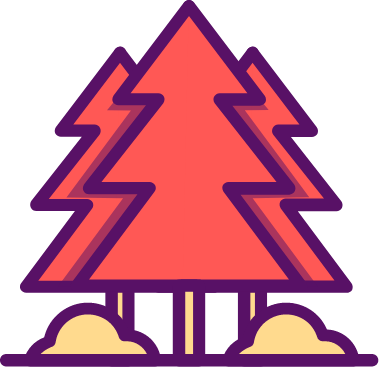


Рисунок 2 – Логотип мобильного приложения «SnowBall»

При первом запуске мобильного приложения пользователя встречает начальный экран (экран авторизации). На данном экране пользователю доступно авторизоваться в приложении или перейти к регистрации аккаунта. В случае если пользователь не помнит свой пароль от учетной записи, на экране предусмотрена функция сброса пароля. Для этого нужно нажать на кнопку «Забыли пароль?». Экран авторизации показан на рисунке 3.

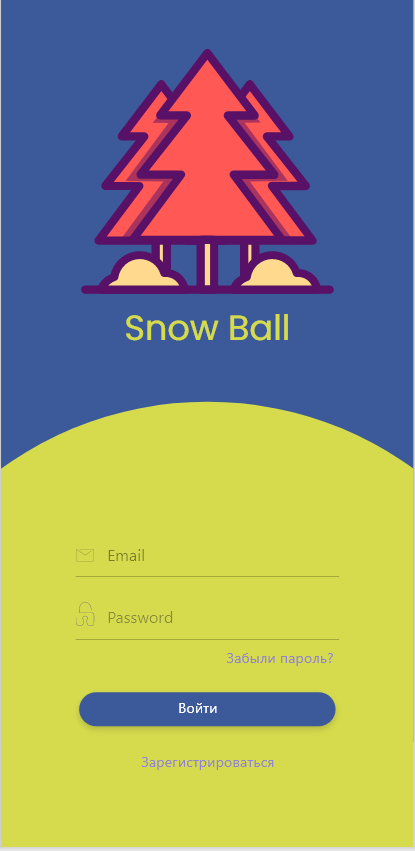


Рисунок 3 – Экран «Авторизация»

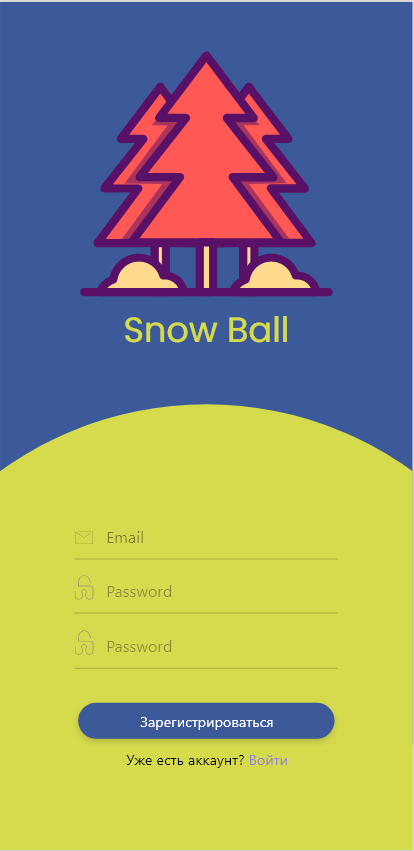
Если пользователь нажмет на кнопку «Регистрация», отобразится экран регистрации аккаунт пользователя (интернет-магазина). Пользователю необходимо предоставить данные его электронной почты, пароля, подтверждение пароля повторным вводом, название магазина и логотип магазина. Входные данные проходят валидацию на соответствие формату электронной почты и пароля. Экран «Регистрация» показан на рисунке 4

Рисунок 4 – Экран «Регистрация»

После регистрации или авторизации пользователя, запускается экран «Сноуборды». В данном экране пользователю выводится список товаров, можно выбрать сноуборд или комплект с ботинками и креплениями. В элементе списка отображается информация о цене на сутки и на тур (3 дня). Для того чтобы добавить в корзину нужный пользователю товар, ему необходимо нажать на кнопку «Добавить» напротив соответствующего товара. Экран «Сноуборды» показан на рисунке 5.

Для переключения между экранами используется нижняя панель навигации (компонент BottomNavigationView).



Рисунок 5 – Экран «Сноуборды»

На экране пользователю выводится список товаров, можно выбрать лыжи или комплект с ботинками и креплениями. В элементе списка отображается информация о цене на сутки и на тур (3 дня). Для того чтобы добавить в корзину нужный пользователю товар, ему необходимо нажать на кнопку «Добавить» напротив соответствующего товара. Экран «Лыжи» показан на рисунке 6.

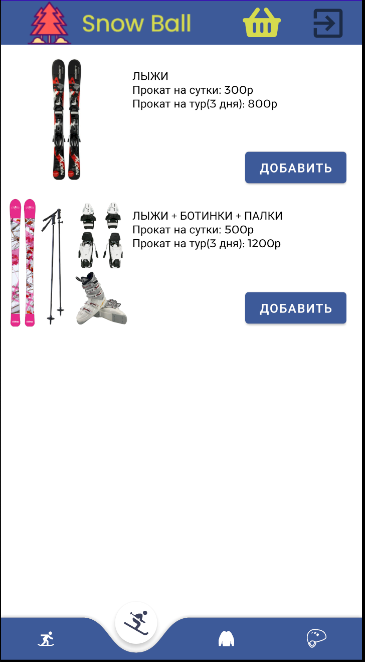


Рисунок 6 – Экран «Лыжи»

На данном экране пользователь может выбрать из списка нужную ему одежду. В элементе списка отображается информация о цене на сутки и на тур (3 дня). Для того чтобы добавить в корзину нужный пользователю товар, ему необходимо нажать на кнопку «Добавить» напротив соответствующего товара. Экран «Одежда» показан на рисунке 7.



Рисунок 7 – Экран «Одежда»

Также пользователь может выбрать из списка нужно ему снаряжение. В элементе списка отображается информация о цене на сутки и на тур (3 дня). Для того чтобы добавить в корзину нужный пользователю товар, ему необходимо нажать на кнопку «Добавить» напротив соответствующего товара. Экран «Снаряжение» показан на рисунке 8.



Рисунок 8 – Экран «Снаряжение»

После того, как пользователь нажмёт на кнопку «Добавить» у него появиться сообщение об успешном добовлении в корзину. Уведомление «Успешно добавлено» показано на рисунке 9.

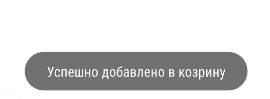
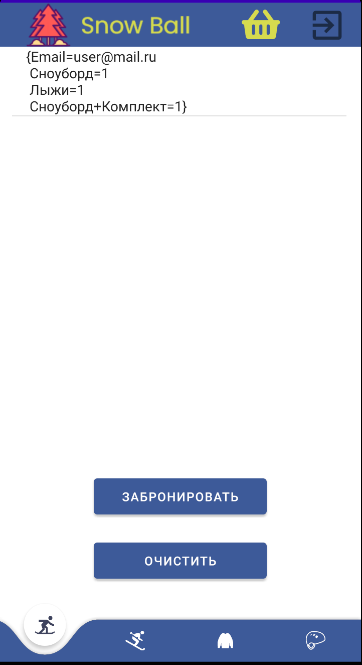


Рисунок 9 – Уведомление «Успешно добавлено»

Когда пользователь закончит добавление товаров в корзину, он должен будет зайти в саму корзину для оформления заказа. На экране «Корзина» пользователь может посмотреть всё, что он добавил в корзину. Экран «Корзина» показан на рисунке 10.

Рисунок 10 – Экран «Корзина»

Как только пользователь убедился, что выбрал всё необходимое, он должен нажать на кнопку «Забронировать» и тогда у него появиться поле «Статус», которое будет равно значению «Забронировано». Экран «Бронирование заказа» показан на рисунке 11.

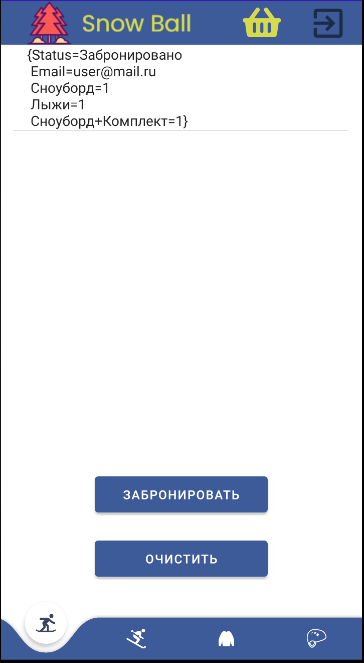


Рисунок 11 – Экран «Бронирование заказа»

После нажатия кнопки «Забронировать», система сообщит о том, что его заказ находится в обработке. Уведомление «Обработка заказа» показано на рисунке 12.



Рисунок 12 – Уведомление «Обработка заказа»

Если пользователь решил изменить свой выбор товаров в корзине, он может нажать на кнопку «Очистить», после чего система очистит его корзину полностью, для того, чтобы он мог заново выбрать товары. После успешной очистки система сообщит пользователю об успешно очистке.

Уведомление «Успешная очистка корзины» показана на рисунке 14.



Рисунок 13 – Уведомление «Успешная очистка корзины»

В панели администратора выводятся вся информация о заказах пользователей, а именно: товары, статус и почта. Экран «Панель администратора» показан на рисунке 13.

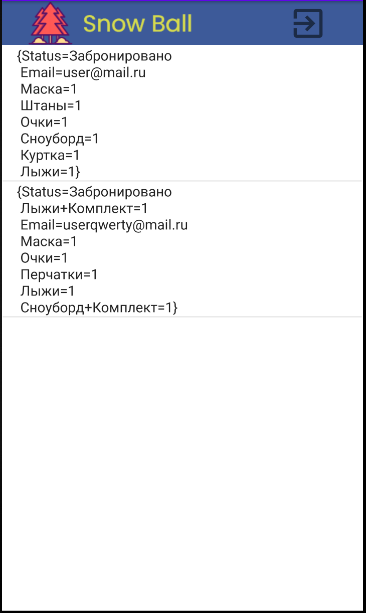


Рисунок 13 – Экран «Панель администратора»

## Разработка форм ввода-вывода информации

На рисунке 14 представлена навигация экранов программного продукта. Экран «Авторизация» отвечает за переход к экрану регистрации и в каталог товаров.

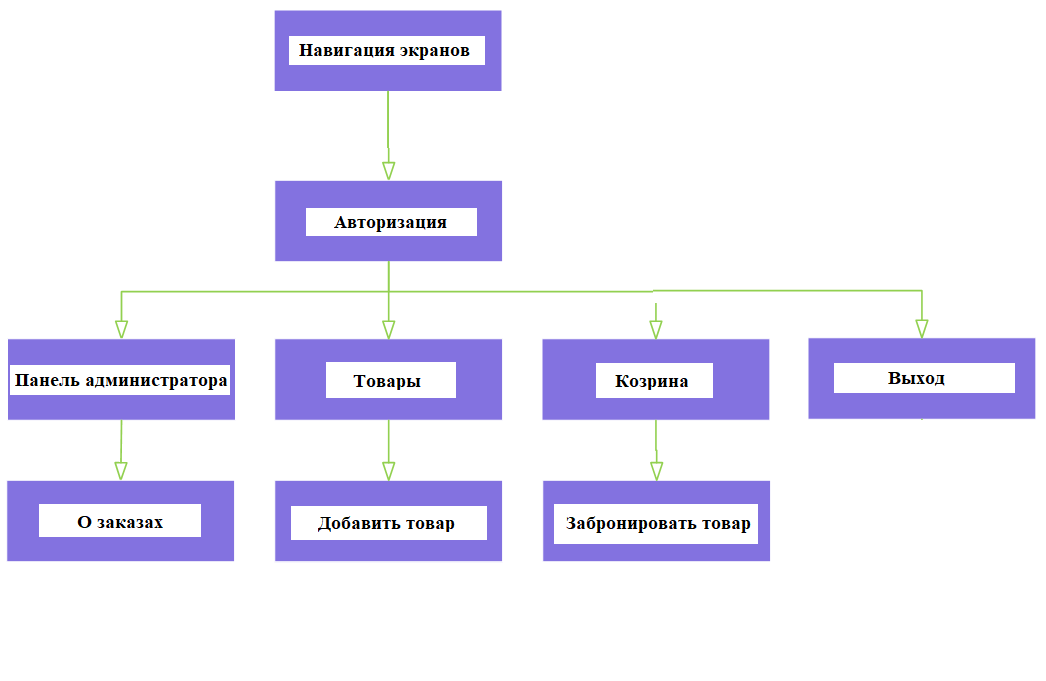


Рисунок 14 – Навигация мобильного приложения

# Выбор методов и разработка основных алгоритмов решения задачи

При проектировании мобильного приложения были разработаны следующие схемы:

1. ER-диаграмма;
2. Алгоритм веб-приложения;
3. Диаграмма прецедентов.

Сущности базы данных описаны в таблице 1.

Таблица 1 — Сущности базы данных

|  |  |
| --- | --- |
| Сущность | Описание сущности |
| Basket | Корзина для товаров |

В соответствии с техническим заданием (Приложение А) для работы использована система управления базами данных Firebase Realtime Database. В Приложении В представлена ER-диаграмма и словарь данных.

Алгоритм мобильного-приложения отображает последовательность операций, используемых в приложении. Алгоритм представлен в Приложении Б.

Диаграмма прецедентов — наглядное представление субъектов и прецедентов со спецификациями (модель предполагаемого поведения информационной системы). Прецеденты удовлетворяют функциональным требованиям за счет предоставления субъекту полезного результата. Диаграмма прецедентов представлена в Приложении Г.

# РАЗРАБОТКА мобильного ПРИЛОЖЕНИЯ

4.1 Описание используемых процедур и библиотечных функций

При разработке программного продукта для реализации поставленных задач были использованы сторонние библиотеки (таблица 2).

Сущности базы данных описаны в таблице 1.

Таблица 2 — Используемые в программе библиотеки.

|  |  |
| --- | --- |
| Наименование библиотеки | Описание библиотеки |
| Firebase Authentication | Firebase Authentication предоставляет серверные службы, простые в использовании SDK и готовые библиотеки пользовательского интерфейса для аутентификации пользователей в мобильном приложении. Он поддерживает аутентификацию с использованием паролей, номеров телефонов, популярных федеративных поставщиков удостоверений, таких как Google, Facebook и Twitter, и других.  Firebase Authentication тесно интегрируется с другими сервисами Firebase и использует отраслевые стандарты, такие как OAuth 2.0 и OpenID Connect. |
| Firebase Realtime Database | Firebase Realtime Database — это база данных, размещенная в облаке. Данные хранятся в формате JSON и синхронизируются в реальном времени с каждым подключенным клиентом. Все пользователи мобильного приложения используют один экземпляр базы данных в реальном времени и автоматически получают обновления с новейшими данными. |
| Firebase Cloud Storage | Firebase Cloud Storage — это мощный, простой и экономичный сервис хранения объектов, созданный для масштабирования Google. SDK Firebase для облачного хранилища добавляют безопасность Google к загрузке и скачиванию файлов для мобильных приложений независимо от качества сети. |
| Material Design | Библиотека для добавления в среду разработки Android Studio компонентов пользовательского интерфейса Material Design. |
| Meow Bottom Navigation | Meow Bottom Navigation представляет собой нижнюю панель навигации, которая позволяет переходить между экранами – пунктами назначения навигации, а также наглядно информирует пользователя о том, на каком экране он находится |

Также при разработке были использованы процедуры, продемонстрированные в таблице 3.

Таблица 3 — Используемые в программе процедуры

|  |  |
| --- | --- |
| Наименование процедуры | Назначение процедуры |
| Авторизация | Выполняется при корректном вводе логина и пароля пользователя, ранее зарегистрированного в приложении. Процедура выполняется на экране «Авторизация». |
| Регистрация | Выполняется при корректном вводе электронной почты, пароля и названия магазина нового пользователя. Процедура выполняется на экране «Регистрация». |
| Проверка на наличие существующей электронной почты | Выполняется при регистрации пользователя с целью предотвращения дублирования учетных записей. |
| Автоматическая авторизация при запуске приложения | Выполняется в случае запуска приложения при условии, что пользователь был ранее авторизирован. |
| Валидации вводимых данных | Выполняется при:   * авторизации пользователя; * регистрации пользователя; * изменении электронной почты пользователя. |
| Изменение данные в Firebase Realtime Database | Выполняется при добавлении, изменении и удалении данных из Firebase Realtime Database. |
| Слушатель изменений в БД | Выполняется в случае изменения каких-либо данных в БД с целью получения актуальной информации в реальном времени. |

Продолжение таблицы 3

|  |  |
| --- | --- |
| Изменение данных в Cloud Storage Firebase | Выполняется при добавлении, изменении и удалении данных из Cloud Storage Firebase. |
| Вывод данных в ListView | Выполняется, когда происходит вывод списка заказов при срабатывании слушателя изменения данных в Firebase Realtime Database |
| Отслеживание изменения ориентации экрана смартфона | Выполняется с целью предотвращения сброса информации на экране приложения при изменении ориентации смартфона. |

4.2 Спецификация приложения

При реализации функциональна программного продукта был использован принцип объектно-ориентированного программирования. Классы и интерфейсы, используемые при разработке мобильного приложения описаны в таблице 4.

Таблица 4 — Используемые в мобильном приложении классы и интерфейсы.

|  |  |
| --- | --- |
| Наименование класса/интерфейса | Описание класса/интерфейса |
| AdminPanel | Класс, содержащий данные о составе заказов всех пользователей. |
| Basket | Класс, содержащий данные о составе заказа пользователя. |
| BlankFragmentSkii | Класс, используемый для вывода списка лыж. |
| BlankFragmentSnowBoard | Класс, используемый для вывода списка сноубордов. |
| Login | Класс, используемый для авторизации пользователей мобильного приложения. |
| Register | Класс, используемый для регистрации пользователей мобильного приложения. |
| BlankFragmentClothes | Класс, используемый для вывода списка одежды. |
| BlankFragmentProtection | Класс, используемый для вывода списка защиты. |

Заключение

Список источников