МИНОБРНАУКИ РОССИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**

**ВЫСШЕГО ОБРАЗОВАНИЯ**

**“ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ”**

Факультет *компьютерных наук*

Кафедра *программирования и информационных технологий*

*Приложение по подбору авиабилетов*

*Курсовая работа*

09.03.02 *Информационные системы и технологии*

*Программная инженерия в информационных системах*

Допущен к защите

Зав. Кафедрой \_\_\_\_\_\_\_\_\_\_\_\_\_\_*С.Д.* Махортов*, к.ф.- м.н., доцент* \_\_.\_\_.20\_\_

Обучающиеся \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*А.Е. Копылова, О.А. Транина 3 курс, д/о*

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*, ст. преподаватель*

Воронеж 2018

# Содержание

[Содержание 2](#_Toc7878797)

[Введение 3](#_Toc7878798)

[1. Постановка задачи 5](#_Toc7878799)

[2. Анализ 7](#_Toc7878800)

[2.1. Анализ предметной области 7](#_Toc7878801)

[2.2. Анализ задач 8](#_Toc7878802)

[2.2.1. Задача поиска авиабилетов 8](#_Toc7878803)

[2.2.2. Задача хранения данных пользователя, приложения и его настроек 8](#_Toc7878804)

[2.3. Анализ средств реализации 9](#_Toc7878805)

[2.4. Диаграммы 10](#_Toc7878806)

[3. Основная часть 19](#_Toc7878807)

[Заключение 19](#_Toc7878808)

[Скелет приложения 20](#_Toc7878809)

[Подсистема поиска авиабилетов 20](#_Toc7878810)

[Подсистема хранения истории поиска (класс SearchHistoryRepository) 22](#_Toc7878811)

[Подсистема хранения закладок 22](#_Toc7878812)

[Подсистема отображения пользовательских интерфейсов. 23](#_Toc7878813)

[Список источников 36](#_Toc7878814)

[Приложения 36](#_Toc7878815)

# Введение

В современном мире мы наблюдаем проникновение информационных технологий практически во все сферы человеческой деятельности, не стала исключением и сфера предоставления услуг по поиску авиабилетов.

Учитывая темп жизни современного человека, время является важнейшим ресурсом. Не удивительно, что с распространением интернета приобрели высокую популярность различные сервисы онлайн-поиска для мобильных устройств. Ведь они позволяют существенно упростить и ускорить процесс получения информации о различных услугах. Очевидно, что использование программного приложения для предоставления информации о перелетах и ценах дает преимущества такие, как:

* экономия времени
* более удобная форма поиска с широким выбором фильтров
* более разнообразные предложения

В области поиска авиабилетов существует большое количество мобильных приложений, работающих на операционной системе Android. Рассмотрим наиболее крупные из них:

* Tutu.ru

Туту.ру (tutu.ru) – российский онлайн сервис путешествий, предоставляющий услуги по поиску и покупки билетов на все виды транспорта, а именно ж/д и авиа перевозки и автобусы. Также предоставляет необходимую информацию путешественникам, например, телефоны авиакомпаний и вокзалов, проезды в аэропорты. Минусом данного приложения является невозможность добавить маршрут или билет в закладки.

* Аэрофлот

Аэрофлот – официальное мобильное приложение ПАО Аэрофлот, предоставляющее возможность поиска, бронирование и покупки авиабилетов. Также предлагает некоторые дополнительные сервисы, такие как регистрация на рейс, онлайн табло и статус рейса, расписание регулярных рейсов. Однако данное приложение предоставляет информацию только о билетах авиакомпании «Аэрофлот».

* Tickets.ru

Tickets.ru – российский онлайн сервис путешествий, предоставляющий услуги по поиску и покупки билетов регулярных авиакомпаний с наиболее актуальными ценами. Недостатком данного приложения является отсутствие возможности сохранить информацию о маршруте и билете для ускорения дальнейшего использования.

При разработке проекта используется каскадная модель, которая включает в себя следующие этапы:

1. Определение требований
2. Проектирование
3. Конструирование (также «реализация» либо «кодирование»)
4. Воплощение
5. Тестирование и отладка (также «верификация»)
6. Инсталляция
7. Поддержка

Также в процессе разработки будет использоваться парадигма объектно-ориентированного программирования.

# Постановка задачи

Целью курсового проекта является создание приложения, работающего на операционной системе Android.

Основную функциональность разрабатываемого приложения отражает диаграмма прецедентов, изображенная на рисунке 1.

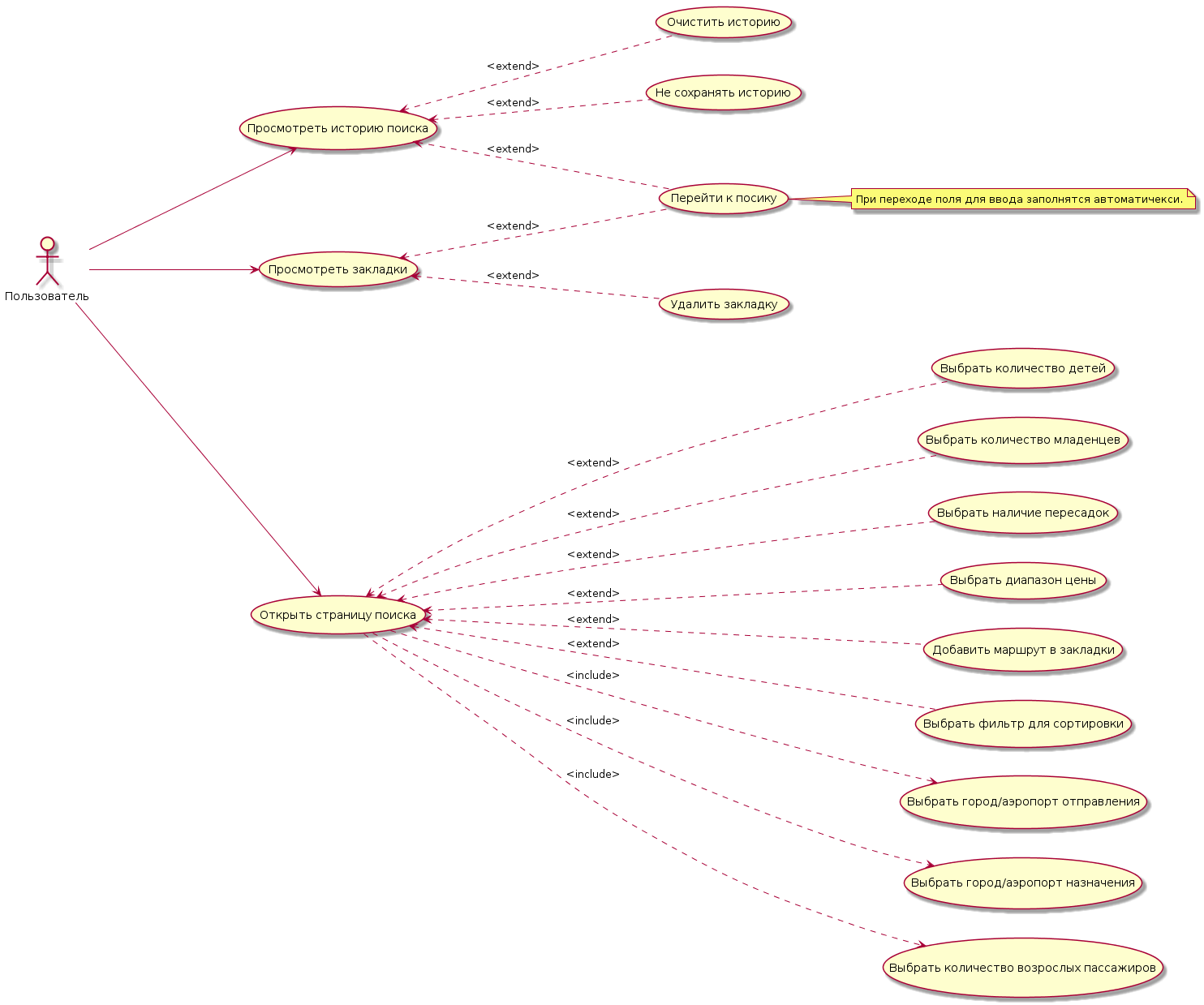


Рисунок 1. Диаграмма прецедентов.

Пользователь обладает следующими возможностями:

* подбор авиабилетов на рейсы в один конец с возможностью добавления информации о количестве пассажиров;
* подбор авиабилетов на рейсы в оба конца с возможностью добавления информации о количестве пассажиров;
* добавление маршрута в закладки;
* просмотр закладок;
* удаление закладки;
* просмотр истории поиска;
* удаление элемента истории поиска;
* очистка истории поиска;
* возможность не сохранять историю поиска**.**

Система должна соответствовать следующим требованиям:

1. Использование нескольких источников данных (API).
2. Подбор не только прямых рейсов, но и рейсов с пересадками.
3. Реализовать все возможности пользователя, описанные выше.

Завершенный проект представляет собой полностью функционирующее Android –приложение, соответствующее требованиям, описанным выше.

# Анализ

## Анализ предметной области

Для поиска авиабилетов минимально необходимой информацией являются следующие данные:

* пункт отправления,
* пункт назначения,
* количество взрослых пассажиров,
* количество детей,
* количество младенцев,
* дата отправления.

Однако часто билеты покупают с обратной дорогой или, как это еще называют, в обе стороны. В этом случае к уже перечисленным данным добавляются данные об обратном билете, в частности дата отправления из пункта назначения.

Также при поиске рейсов из пункта назначения в пункт прибытия необходимо учитывать не только прямые рейсы, но и рейсы с пересадками, то есть такие, у которых есть промежуточные точки остановок в пути. Однако не для всех пассажиров такие рейсы являются приемлемыми, так как они существенно снижают комфорт путешествия, добавляя к времени в дороге время на саму пересадку. Для таких случаев следует предоставить возможность пользователю приложения убрать такие рейсы из итоговых результатов поиска.

По причине того, что результатом поиска, как правило, является длинный список разнообразных вариантов рейсов в пункт назначения, пользователю следует предоставить возможность сортировки результатов по следующим критериям:

* по убыванию цены;
* по возрастанию цены;
* по времени в дороге (от самого долгого путешествия к самому короткому);
* по времени в дороге (от самого короткого путешествия к самому долгому);
* по количеству пересадок (от большего к меньшему);
* по количеству пересадок (от меньшего к большему).

## Анализ задач

### Задача поиска авиабилетов

Рассмотрим задачу поиска авиабилетов. Данная задача включает в себя следующие этапы:

1. Проверка данных, введенных пользователем. Если данные введены не корректно, то пользователю предлагают ввести их повторно.
2. Формирование и отправление запроса с данными на сторонний ресурс (API сервер).
3. Обработка ответа, если таковой получен. Если ответ не получен или произошла какая-либо ошибка, то пользователю выводится сообщение об этом и предлагается ввести данные для поиска повторно.
4. Отображение итоговых результатов поиска пользователю.

### Задача хранения данных пользователя, приложения и его настроек

Рассмотрим задачу хранения данных пользователя, приложения и его настроек. Данную задачу можно подразделить на две подзадачи, а именно:

* хранение закладок,
* хранение истории поиска.

#### Подзадача хранения закладок

Рассмотрим подзадачу хранения закладок. Закладка содержит в себе следующую информацию:

* пункт отправления;
* пункт назначения;
* количество взрослых пассажиров;
* количество детей;
* количество младенцев;
* наличие пересадок;
* тип путешествия (в один конец или с обратной дорогой);
* класс обслуживания (эконом или бизнес);

Для выполнения данной подзадачи следует реализовать следующие возможности пользователя:

1. Сохранение выбранного маршрута в закладки.
2. Просмотр всех закладок.
3. Удаление закладки.
4. Переход с закладки к форме поиска с заполненными полями, соответствующими информации закладки.

#### Подзадача хранения истории поиска

Рассмотрим подзадачу хранения истории поиска. Отдельным элементом истории поиска является совокупность информации, введенной пользователем для поиска рейсов. Это значит, что в данном элементе содержится информация о следующем:

* пункт отправления,
* пункт назначения,
* количество взрослых пассажиров,
* количество детей,
* количество младенцев,
* дата отправления.
* дата отправления из пункта назначения (если билеты в оба конца)
* наличие пересадок
* тип путешествия (в один конец или с обратной дорогой);
* класс обслуживания (эконом или бизнес);

Также данная подзадача включает в себя задачи по реализации возможности пользователя:

* просматривать историю поиска;
* удалять отдельный элемент истории поиска;
* очистить всю историю поиска;
* не сохранять историю поиска;
* переход с элемента истории поиска к форме поиска с заполненными полями, соответствующими информации закладки.

## Анализ средств реализации

В качестве средств реализации системы поиска авиабилетов были выбраны следующие технологии:

1. Android SDK – средство разработки мобильных приложений для операционной системы Android. Чертой, отличающей от других средств разработки, является наличие широких функциональных возможностей, позволяющих запускать тестирование и отладку исходных кодов, оценивать работу приложения в режиме совместимости с различными версиями ОС Android.
2. В качестве СУБД была выбрана SQLite, так как закладки было решено хранить во внутренней базе данных устройства, работающего на операционной системе Android.

Для реализации отдельных подзадач были выбраны следующие библиотеки и средства:

* Протокол прикладного уровня HTTPS и библиотека Retrofit 2 Android для взаимодействия с внешними API серверами.
* Библиотека Glide для асинхронной загрузки изображений из сети интернет.
* Библиотека Room для взаимодействие с базой данных.

## Диаграммы

Для описания работы системы были составлены следующие диаграммы:

* Диаграмма активности изображена на рисунке 2.
* Диаграмма коммуникаций изображена на рисунке 3.
* Диаграмма последовательностей изображена на рисунке 4.
* Диаграмма состояний изображена на рисунке 5.
* Диаграммы классов изображены на рисунках 6-10.
* Диаграмма объектов изображена на рисунке 11.
* Диаграмма развертывания изображена на рисунке 12.

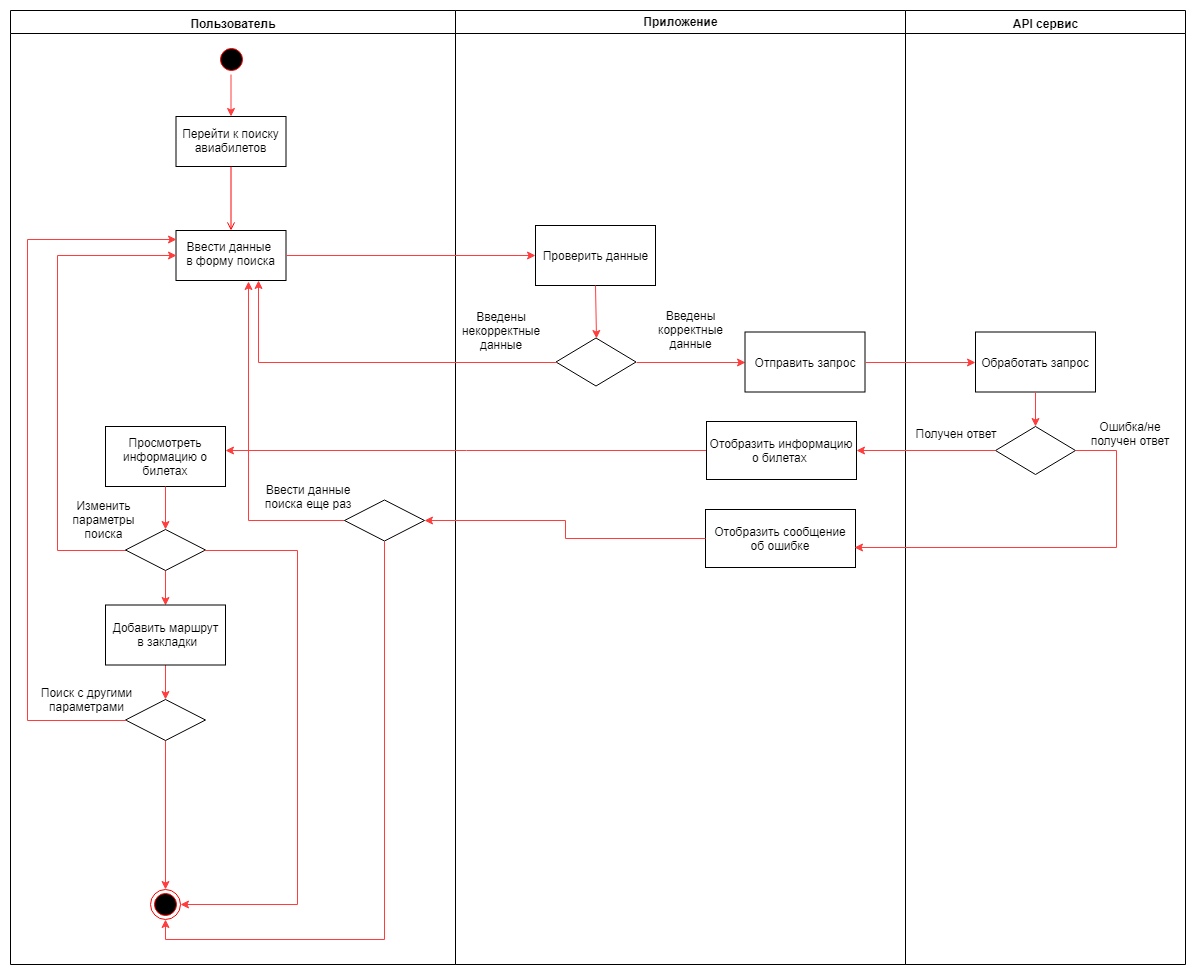


Рисунок 2. Диаграмма активности.

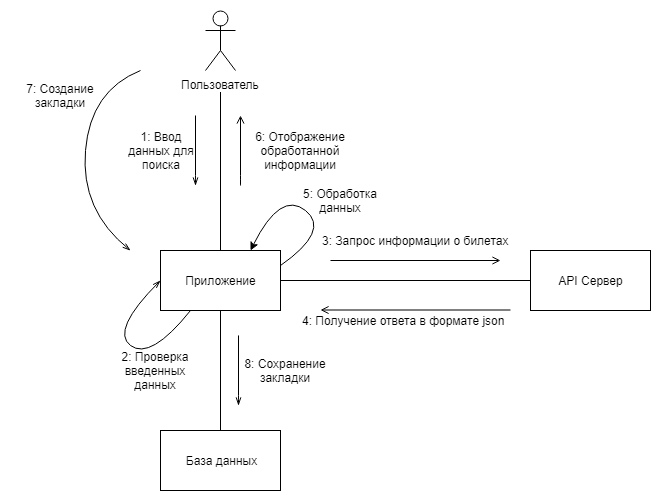


Рисунок 3. Диаграмма коммуникаций.

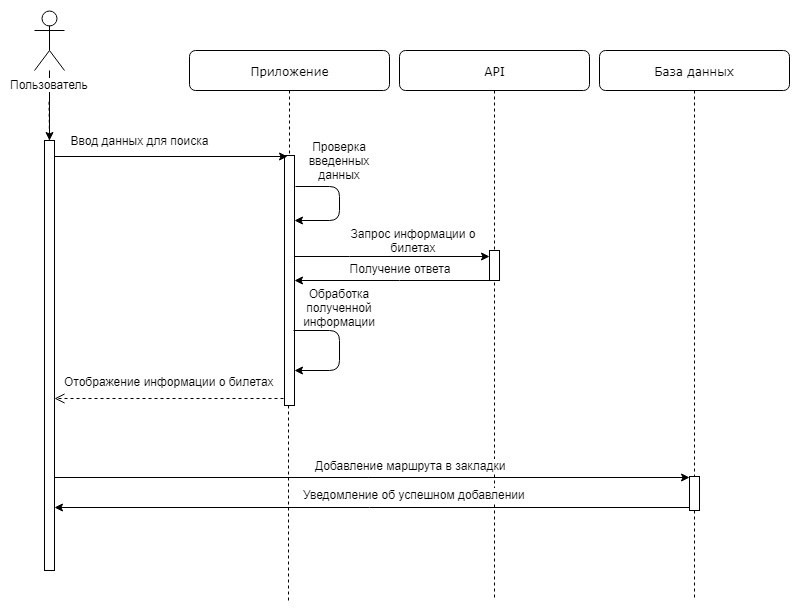


Рисунок 4. Диаграмма поседовательностей.

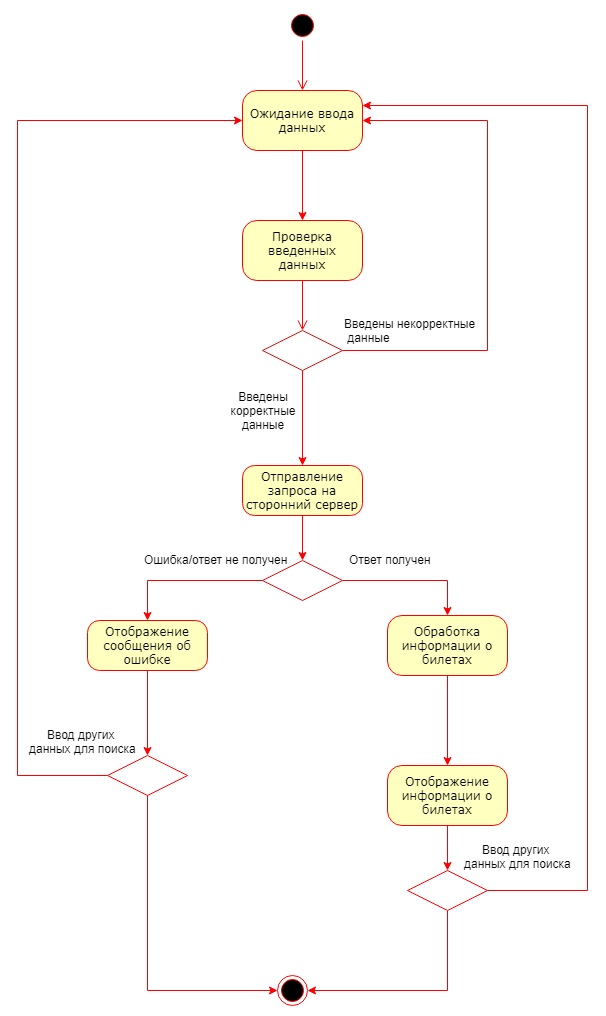


Рисунок 5. Диаграмма состояний.

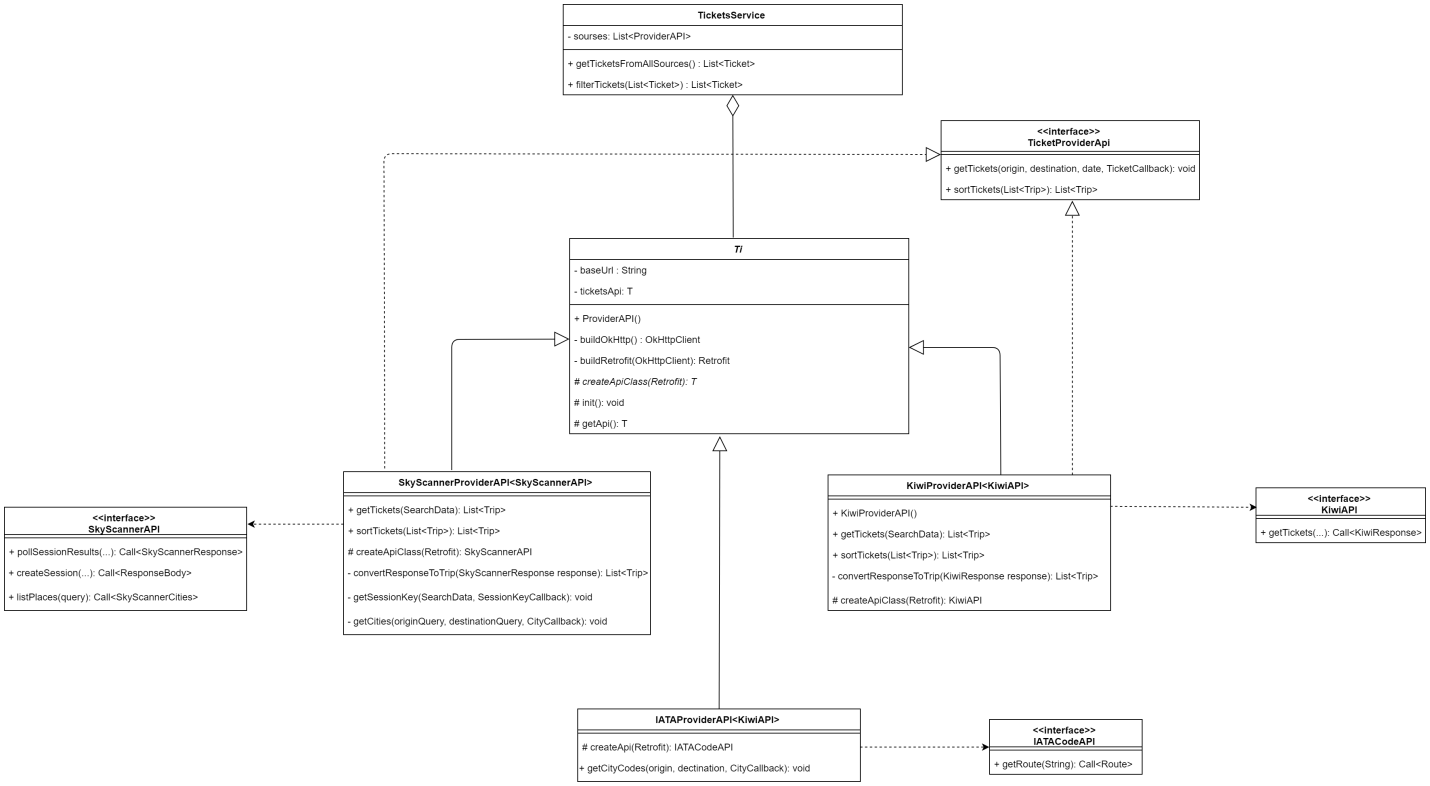


Рисунок 6. Диаграмма классов модуль поиска авиабилетов.

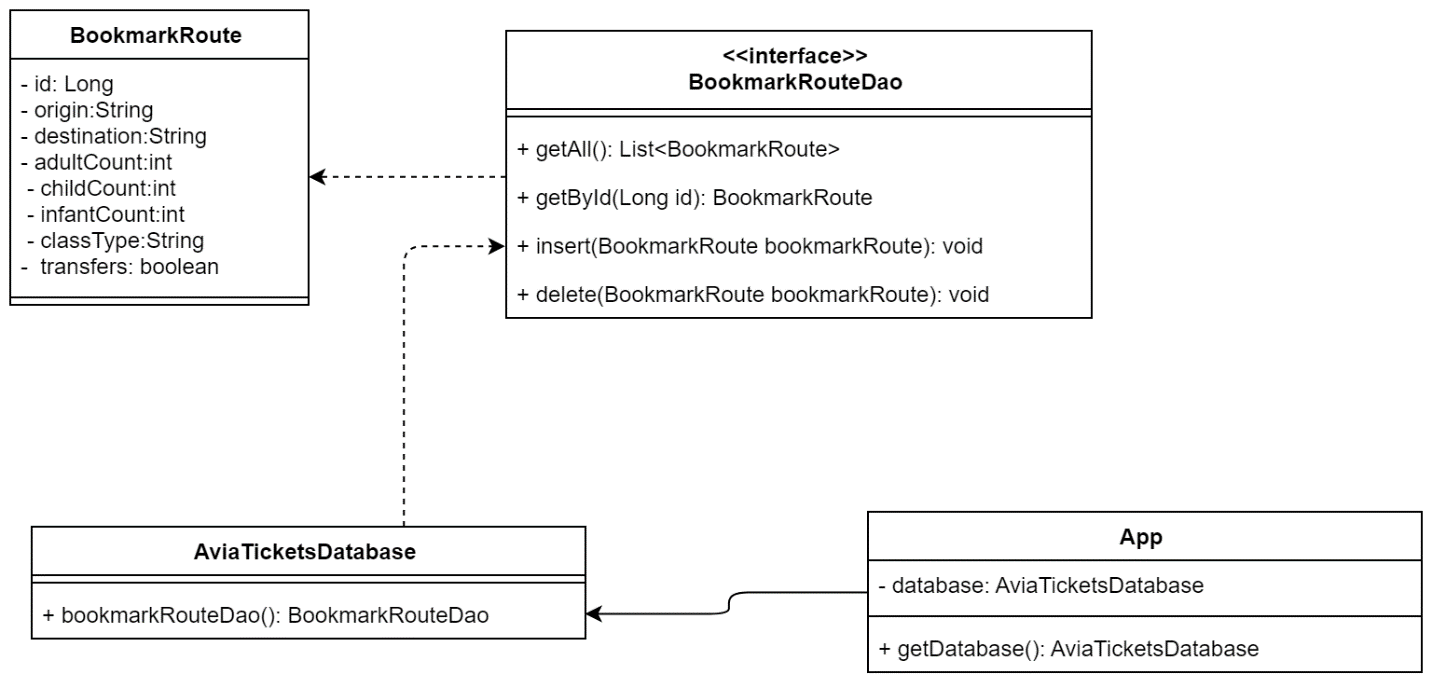


Рисунок 7. Диаграмма классов общения с базой данных.

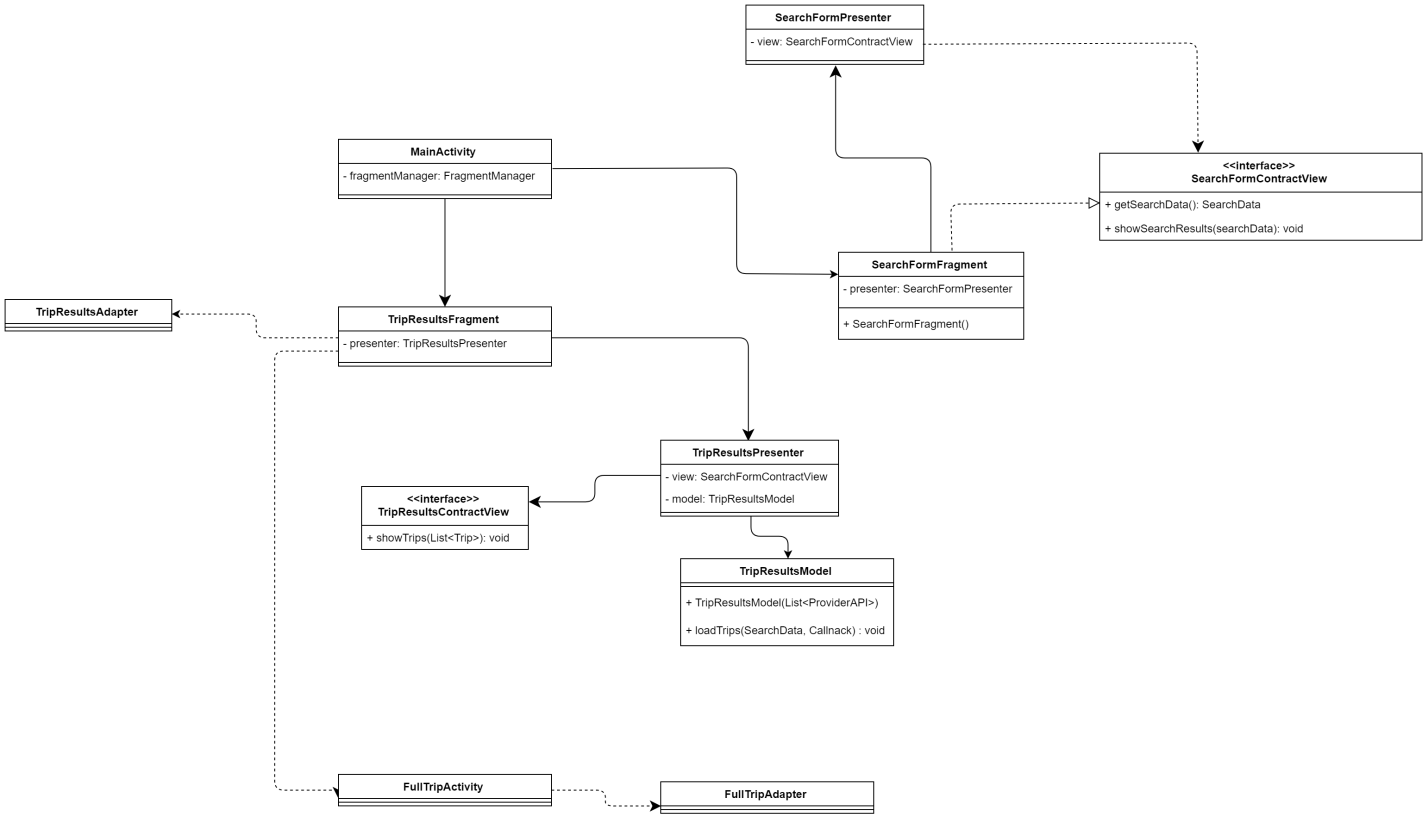


Рисунок 8. Диаграмма классов пользовательского интерфейса.

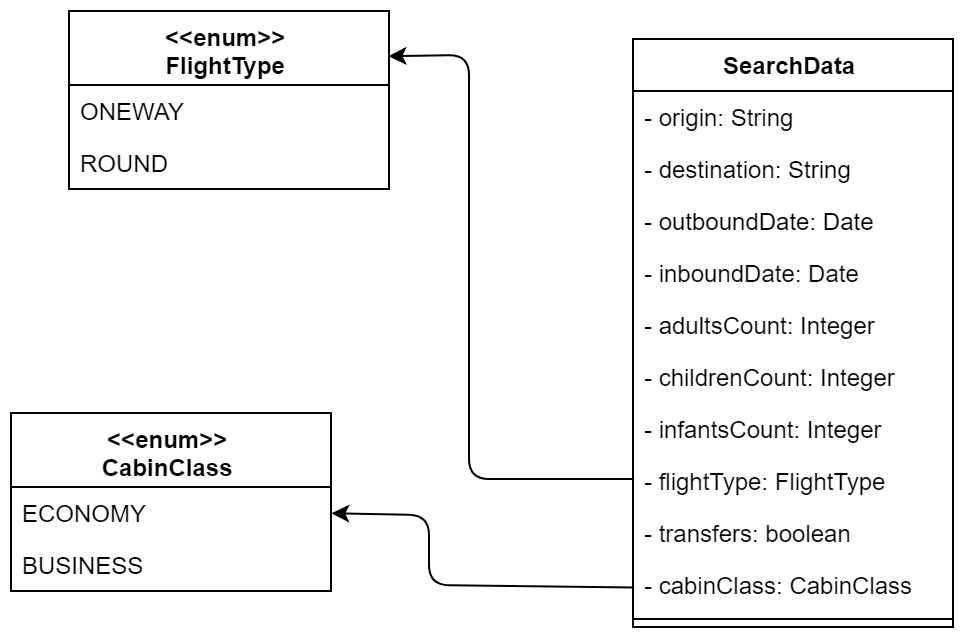


Рисунок 9. Диаграмма классов чтения из пользовательского интерфейса.

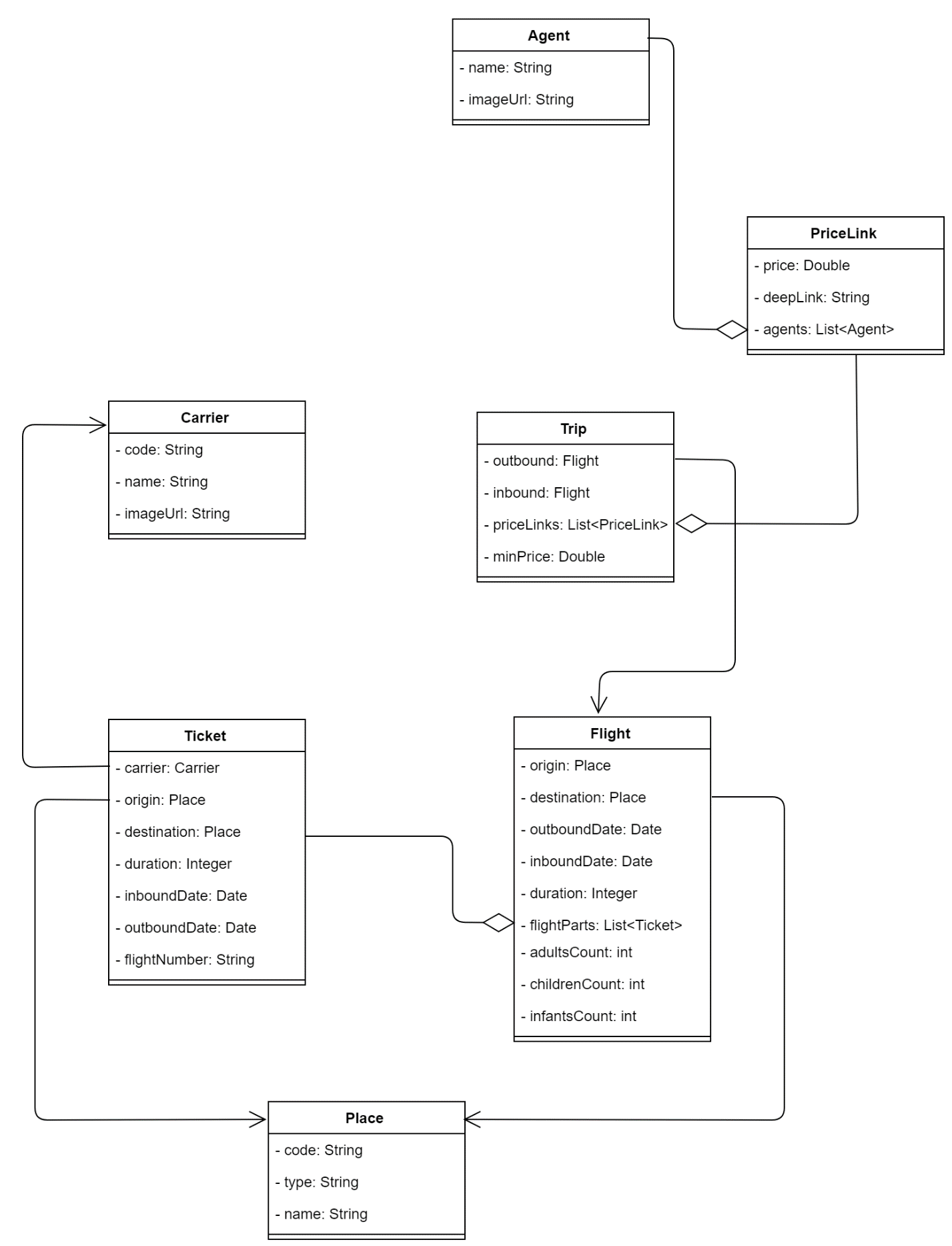


Рисунок 10. Диаграмма классов полученных билетов.

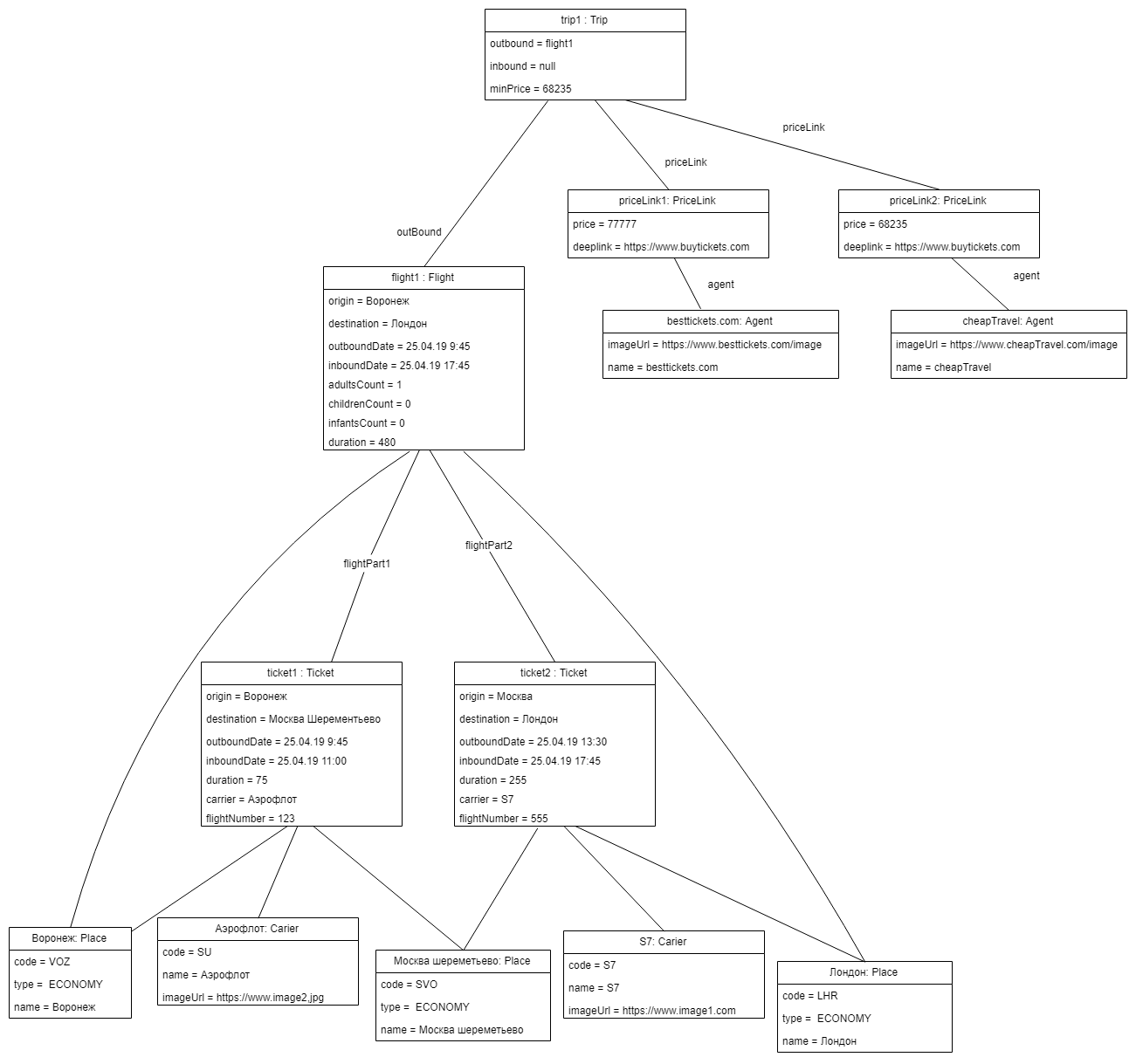


Рисунок 11. Диаграмма объектов полученных билетов.

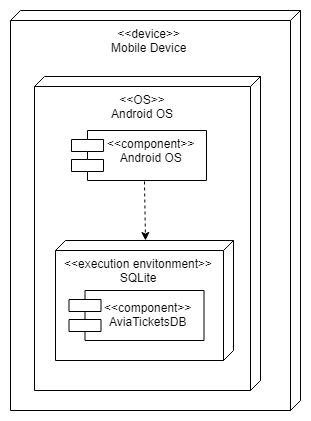


Рисунок 12. Диаграмма развертывания.

# Основная часть

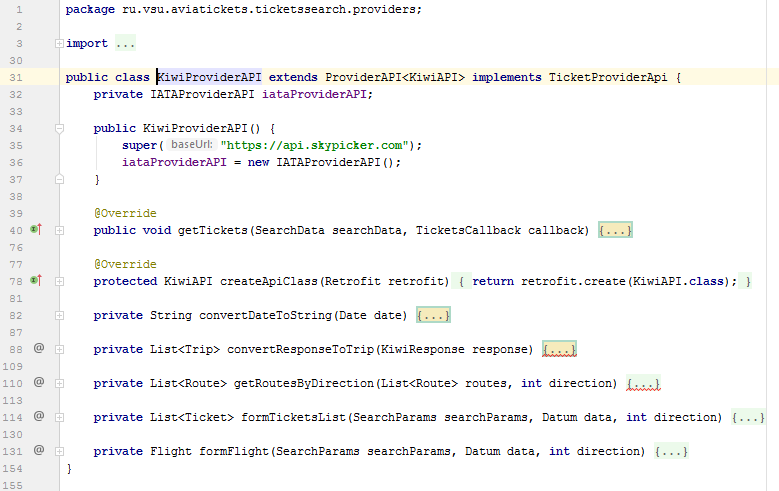
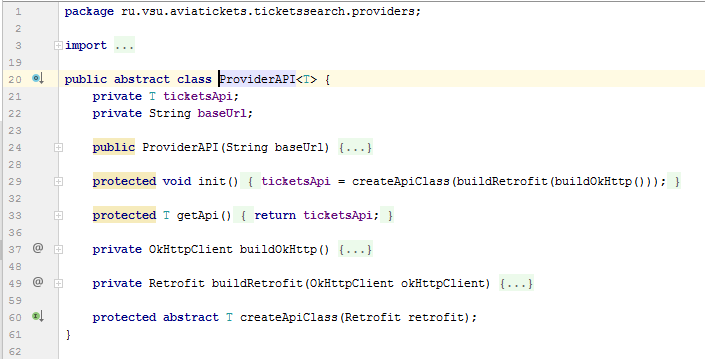
# Заключение

# Скелет приложения

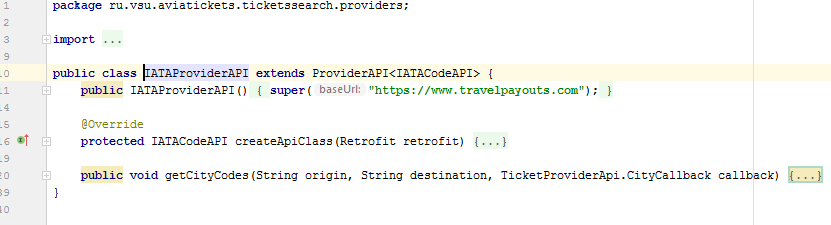
В данном разделе содержатся классы приложения, отвечающие за его функциональность.

## Подсистема поиска авиабилетов

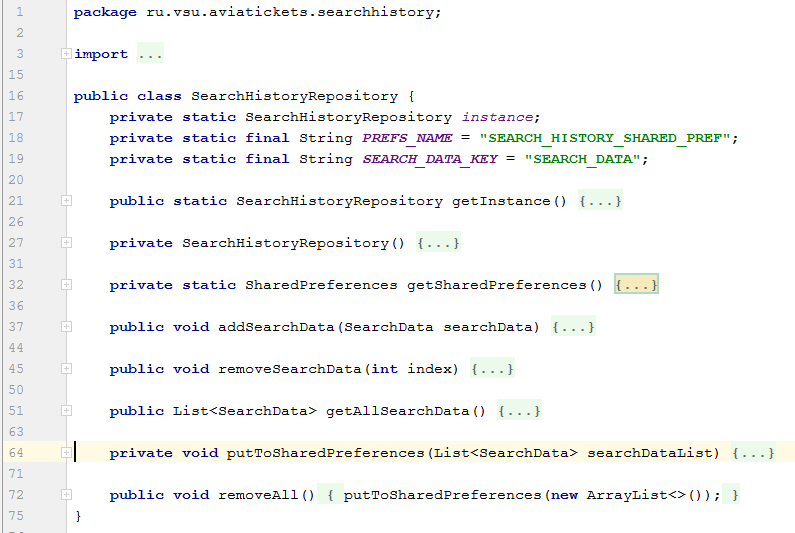
Классы, отвечающие за запрос к сторонним источникам и обрабатывающие ответ: (Отсутствуют класс, описывающие объекты, и перечисления для упрощения чтения)





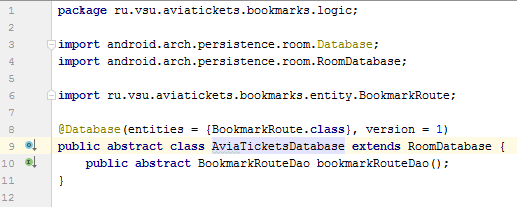


## Подсистема хранения истории поиска (класс SearchHistoryRepository)

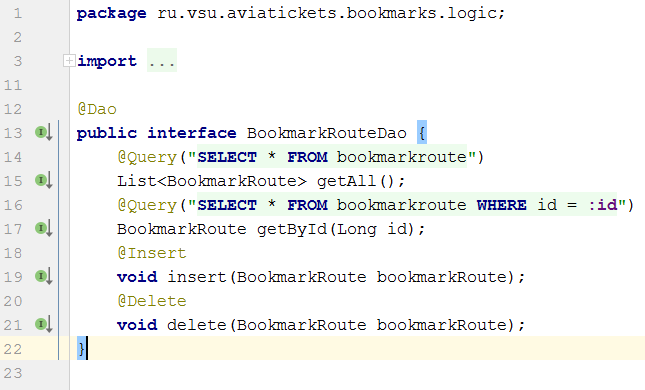


## Подсистема хранения закладок

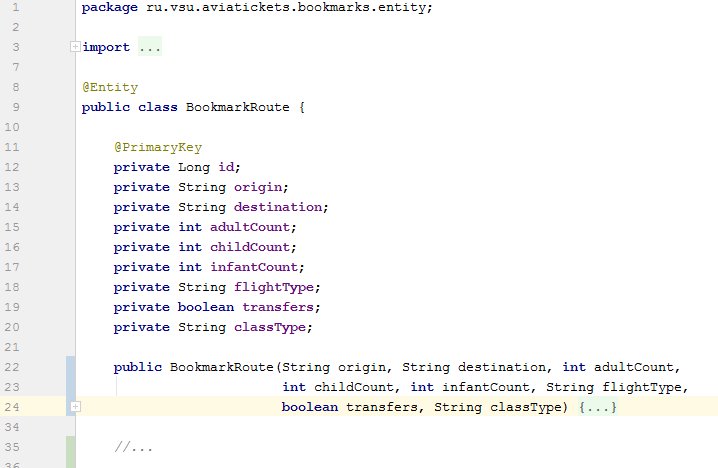
Класс AviaTicketsDatabase отвечает за связь с базой данных



Класс BookmarkRouteDao, отвечающий за запросы к базе данных

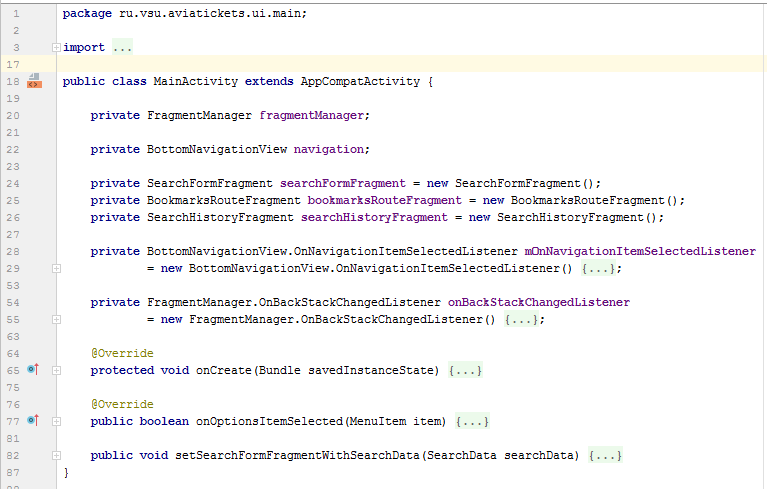


Класс BookmarkRoute, описывающий закладку.



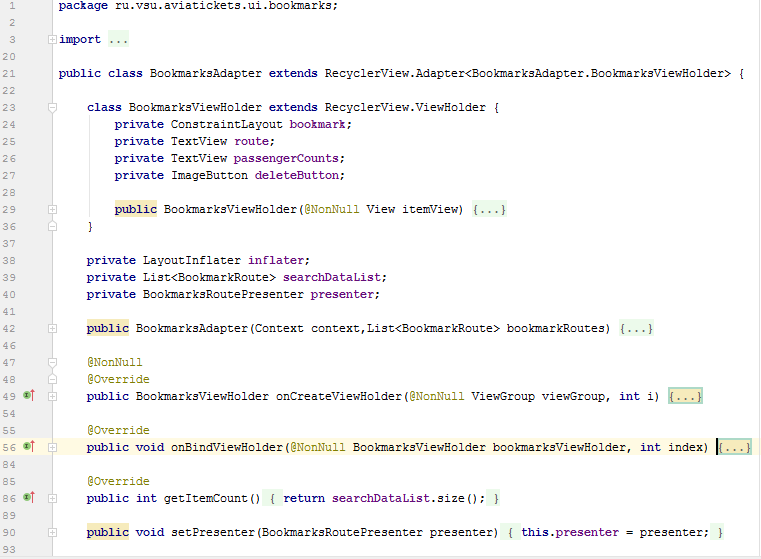
## Подсистема отображения пользовательских интерфейсов.

Основная страница MainActivity

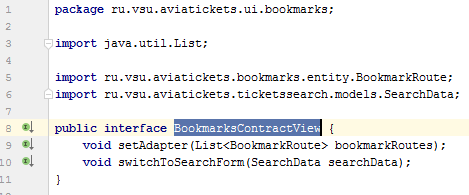


**Отображение страницы с закладками включает в себя классы:**

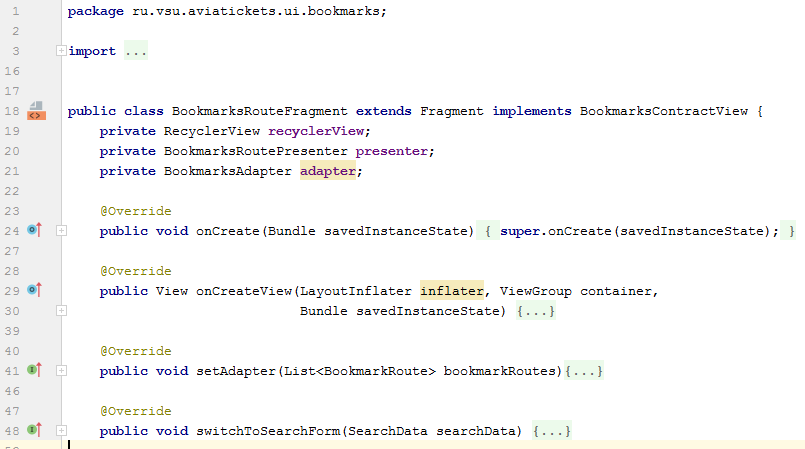
BookmarksAdapter (необходим для отображения списка)



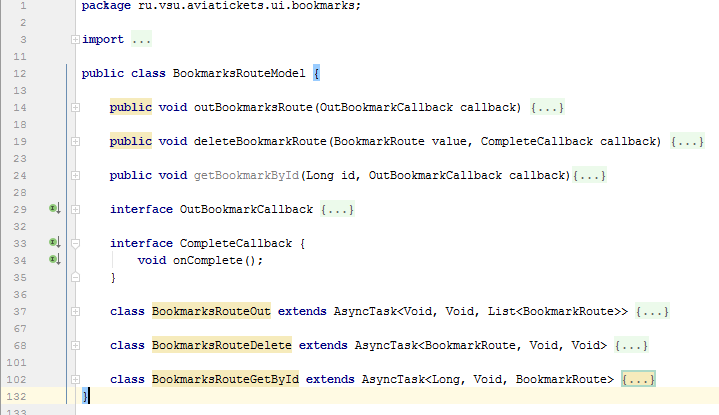
Интерфейс BookmarksContractView для связи с фрагментом (часть авктивити)



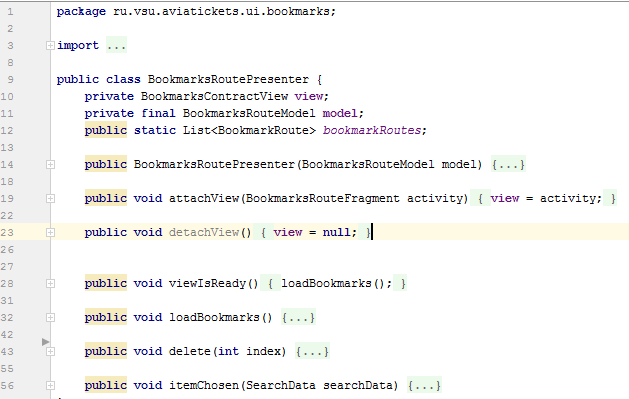
Класс BookmarksRouteFragment для отображения



Класс BookmarksRouteModel отвечающий за логику. Здесь классы, наследующиеся от AsyncTask, выполняются в отдельном потоке для того, чтобы тяжеловесность процесса не влияло на отображение интерфейса.

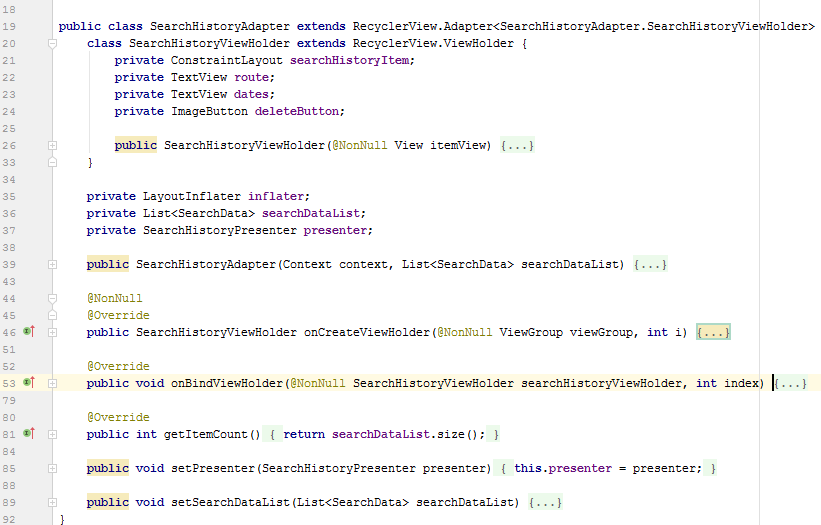


Класс BookmarksRoutePresenter – взаимосвязь отображения и логики.

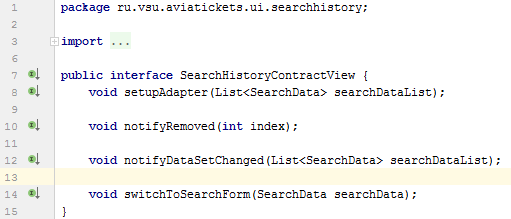


**Отображение страницы истории поиска включает в себя классы** (аналогично классам отображения страницы с закладками**)**

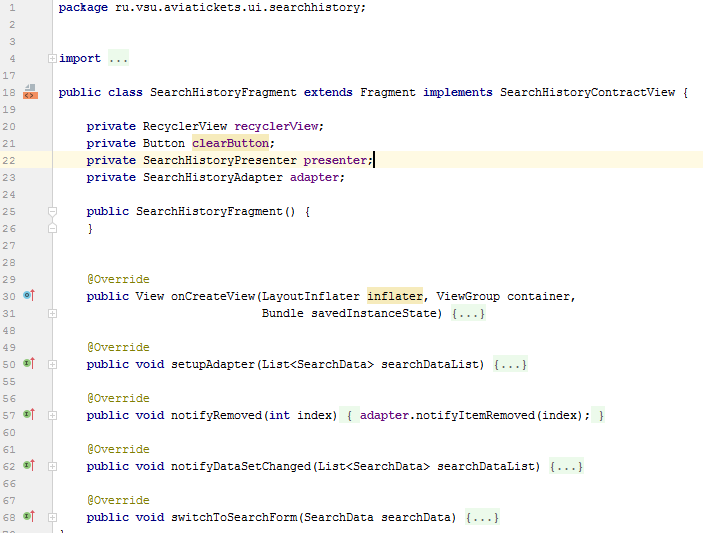
**SearchHistoryAdapter**



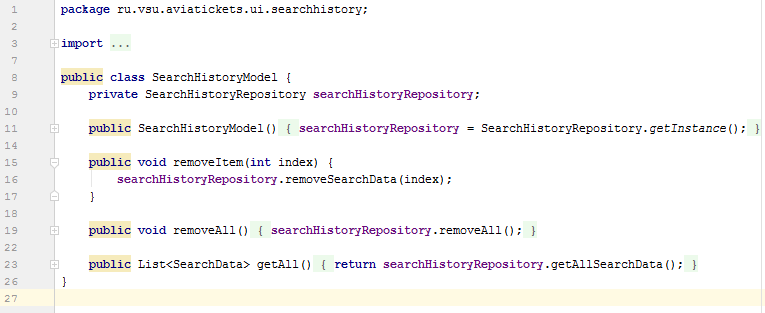
SearchHistoryContractView



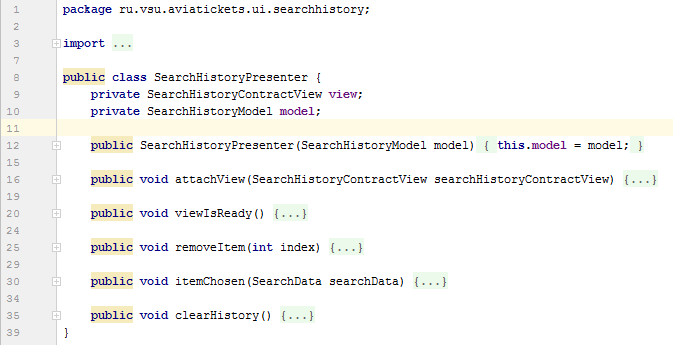
SearchHistoryFragment



SearchHistoryModel

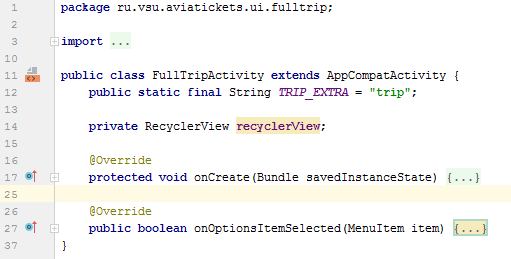


SearchHistoryPresenter

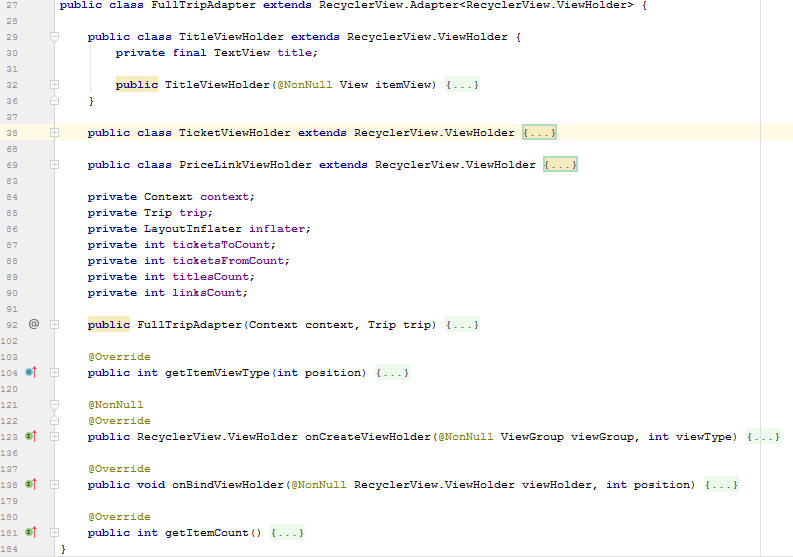


**Отображение выбранного рейса включает в себя классы**

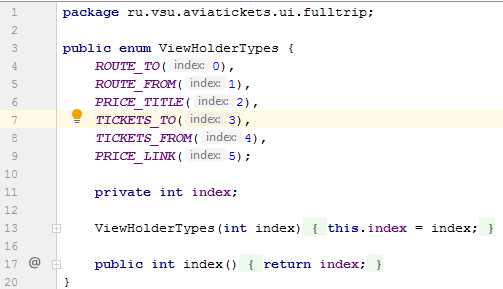
FullTripActivity



FullTripAapter

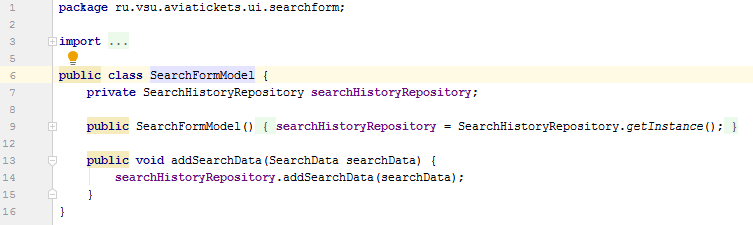


Перечисление ViewHolderTypes

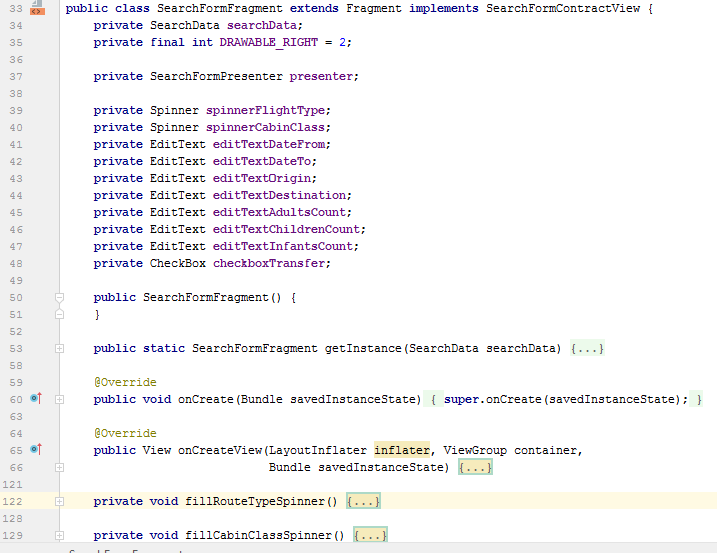


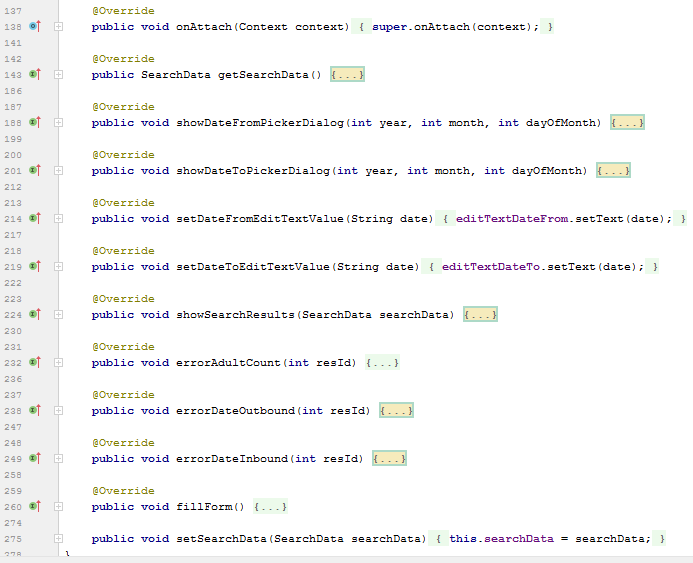
Отображение формы поиска: ( аналогично странице отображения закладок)

SearchFormModel

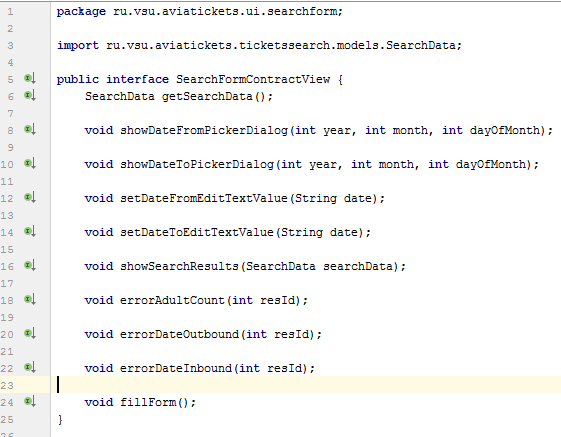


SearchFormFragment

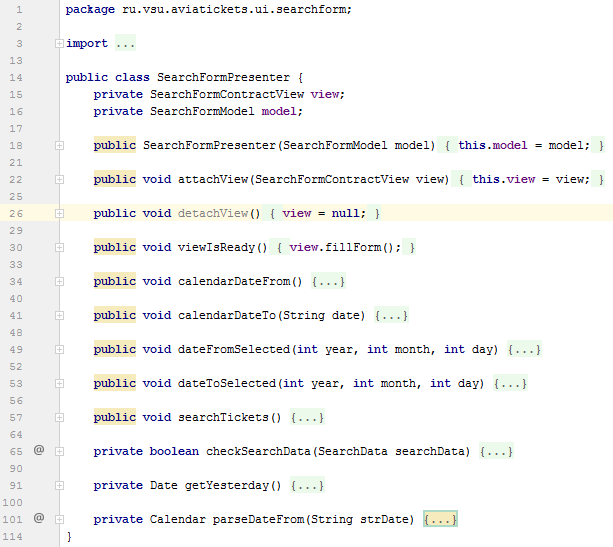




SearchFormContractView

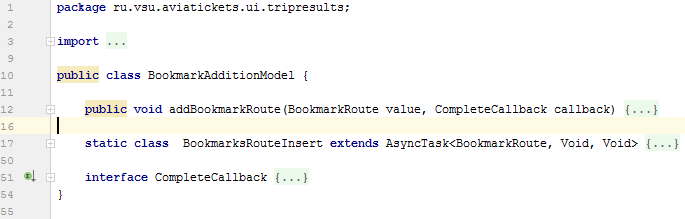


SearchFormPresenter

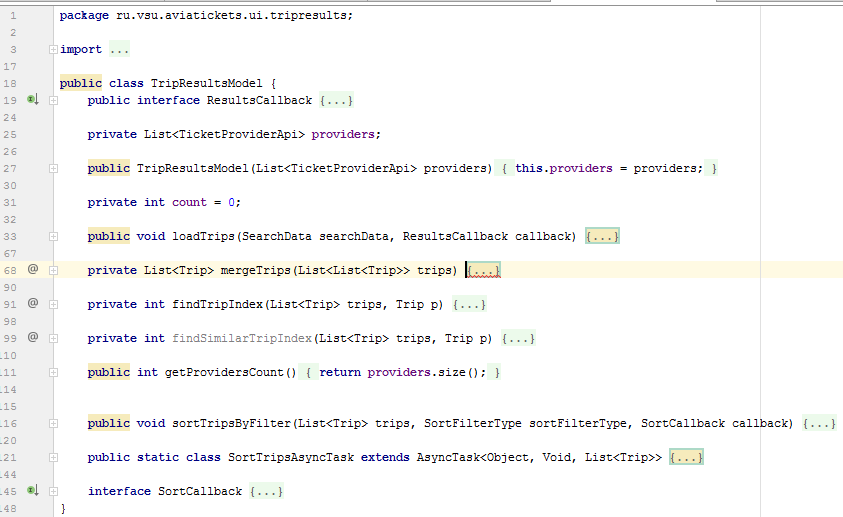


Отображение страницы результатов поиска: (аналогично отображению страницы закладок)

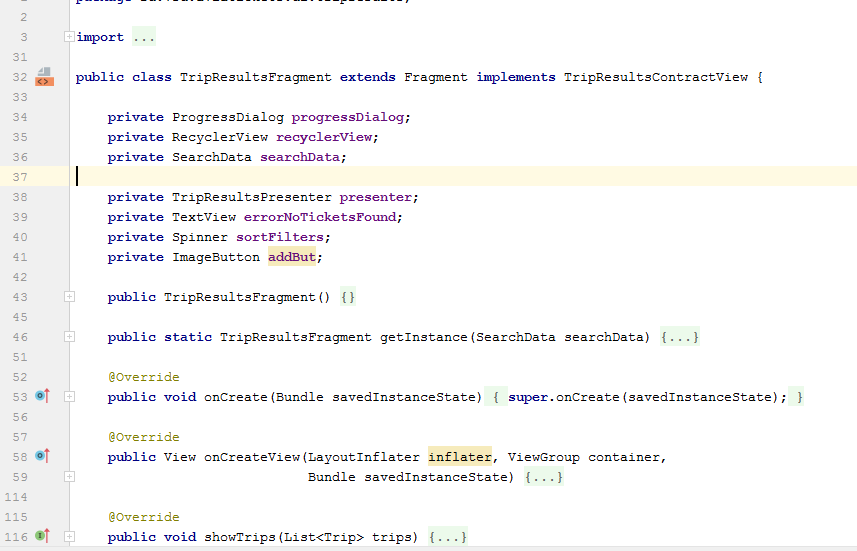
BookmarkAdditionModel

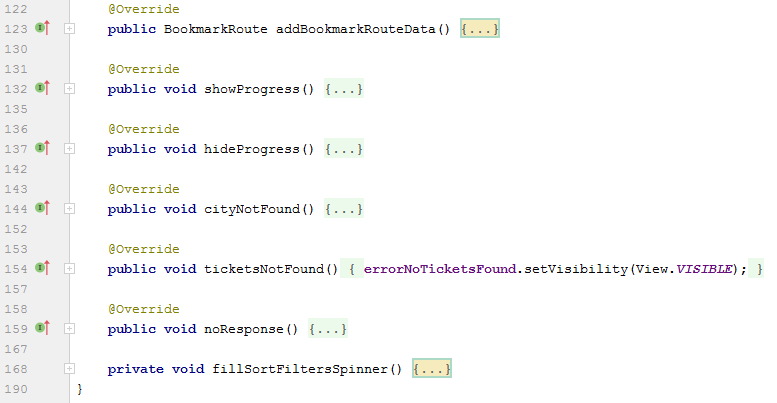


TripResultsModel

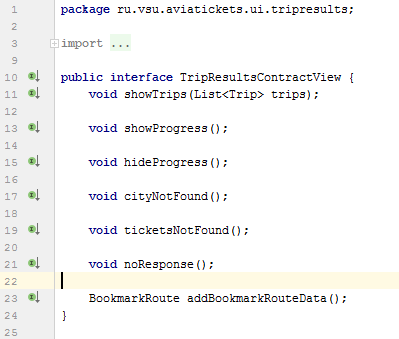


TripResultsFragment

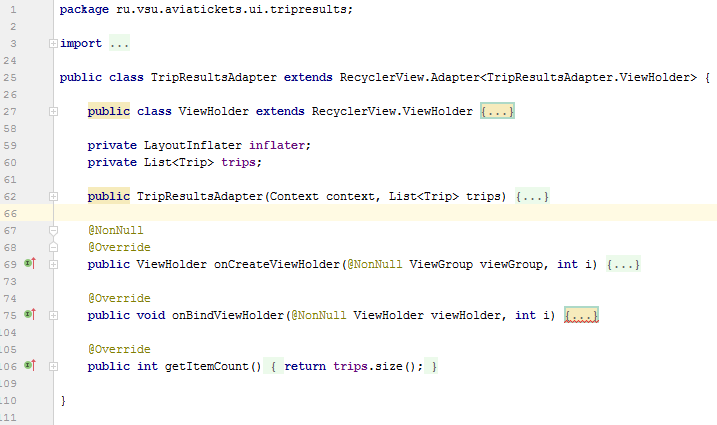




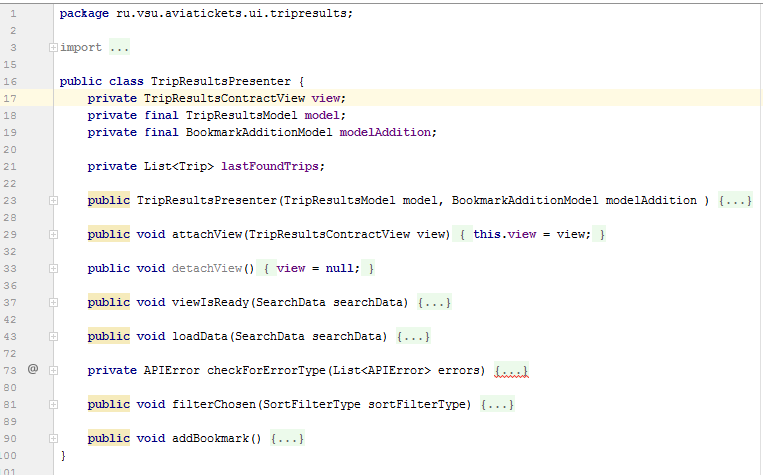
TripResultsContractView



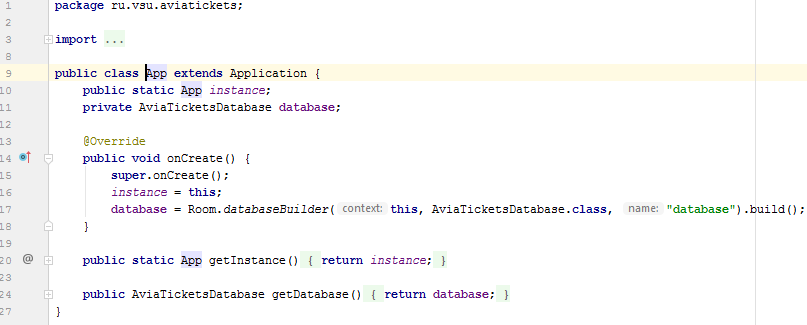
TripResultsAdapter



TripResultsPresenter



Также в некоторых классах используется класс App для реализации паттерна одиночка.



# Список источников

# Приложения