!!!!!!!!!!!!!!!!!!!!!!!!

Подписать все картинки и таблицы

Реферат

Записка 34 с., 3 ч., 24 рис., 6 табл., 4 источника, 2 прил.

Результатом выполнения работы является программа «Энциклопедия». Программа предназначена для ознакомления с разнообразными объектами космоса и закрепления полученной информации в игровой форме.

Цель работы – разработка программы предоставляющей информацию по космосу и возможность закрепления материала.

Возможности программы:

* Вывод информации по объектом звездного неба
* Игра вопрос-ответ
* Просмотр статистики пользователя

Программой может использоваться на смартфонах под управлением ОС Андроид, версия которой выше версии 2.3.3.

**Содержание**

Введение 5

1 Анализ требований и уточнение спецификаций 6

1.1 Исследование предметной области 6

1.2 Выбор технологии, языка и среды разработки 7

1.3Построение диаграммы вариантов использования 8

2 Проектирование структуры и компонентов программного продукта 10

2.1 Проектирование интерфейса программного продукта 10

2.1.1 Проектирование диаграммы классов интерфейсной части 10

2.1.2 Проектирование структуры меню 12

2.1.3 Построение графа состояния интерфейса 13

2.1.4 Выбор типа интерфейса и формы диалога 14

2.1.5 Разработка форм ввода-вывода информации 15

2.2 Разработка алгоритмов методов 16

3 Выбор стратегии тестирования и разработка тестов 21

Заключение 24

Литература 25

Приложение А. Техническое задание.

Приложение Б. Руководство пользователя.

Введение

Смотря на звездное небо, хочется ориентироваться в созвездиях, знать названия, знать расположение и быстро находить их. Для этого надо выучить все созвездия, но на это нужно потратить немало времени, которое так не хватает в современном мире. В реальной жизни не каждый может позволить себе потратить десяток вечеров для изучения всего материала.

Разработанное приложение написано под ОС Андроид и позволяет изучать созвездии, планеты и звезды в любом месте в любое время.

Существующие аналоги, такие как «SkyMap», «StarWalk» или «StarTracker», дают пользователю ограниченный функционал, либо карту неба, либо названия созвездий, либо просто теорию по созвездиям и другим объектам космоса. Еще есть интернет ресурс Google Sky, но для использования нужен хороший интернет.

Моё приложение хранит информацию во встроенной ОС базе данных. Таким образом можно быстро получить доступ к данным по хранящимся там темам (созвездия, планеты и т.д.). А также пользователь имеет возможность проверить или закрепить свои знания по созвездиям и планетам.

НАДО НАПИСАТЬ НА СТРАНИЦУ см черновик

**1. Анализ требований и уточнение спецификаций**

**1.1 Исследование предметной области**

В данной работе поставлена задача создать удобное и быстрое приложение, которым можно воспользоваться в любое время при наличии смартфона на базе ОС Андроид.

Многие люди любят проводить свободное время, наблюдая за звездами, складывая из них определенные рисунки, находя созвездия. Еще в древности многие люди видели в небе картинки, которые напоминали людей, животных или какие-либо предметы. Так возникла, например, Большая Медведица, которая на сегодняшний день является наиболее популярным и широко известным созвездием. На сегодняшний день нам известно 88 созвездий, в состав которых входят все звезды, которые мы видим на небе.

Очень часто пользователю интересно название и расположение созвездий и звезд. Надо предоставить возможность пользователю найти созвездие на небе по картинке, словесному описанию положения относительно других объектов неба. Не все приложения могут предоставить полную информацию по этой теме.

Помимо этого у каждого созвездия есть интересная история о том почему они так называются, история открытия.

Все созвездия состоят из звезд, которые получили свои имена еще в древности. Эти звезды светятся по-разному из-за того, что имеют разную структуру и относятся к разным типам. На данный момент существую более 14 типов звезд. О которых пользователю будет интересно знать.

**1.2 Выбор технологии, языка и среды разработки**

Технологией программирования называют совокупность методов и средств, используемых в процессе разработки программного обеспечения. Для разработки моей программы было принято решение использовать спиральную схему разработки из трех известных схем разработки (а именно схема с промежуточным контролем и каскадная схема разработки), потому что позволяет создавать программу поэтапно и последовательно.

Для написания программы необходимо было выбрать язык программирования и среду разработки. К данной программе были сформулировала следующие функциональные требования:

* Поддержка объектного подхода;
* Событийное программирование;
* Модульность, с использованием правил вертикального управления.

При разработке интерфейса пользователя будет использован объектный подход в совокупности с технологией событийного программирования.

Для создания программного продукта был выбран язык высокого уровня Java. Язык прост в освоении и не сильно отличается от языка С++. В качестве среды разработки использовал Android Studio. Android Studio много функциональная среда предназначенная для разработки мобильных приложений на базе ОС Андроид. Среда разработки имеет удобный инструмент для работы с SDK и предоставляет быстрый доступ к созданию и управлению AVD, что ускоряет тестирование и проверку работоспособности приложения на разных этапах разработки.

**1.3 Построение диаграммы вариантов использования**

При проектировании интерфейса необходимо учесть специфику и предназначение программного продукта. В таблицах N-N представлено варианты использования в кроткой форме.

|  |  |
| --- | --- |
| Название варианта | **Просмотр данных об объектах** |
| Цель | Получение сведений по объектам звездного неба |
| Действующие лица | Пользователь |
| Краткое описание | Пользователь выбирает режим «Энциклопеди», затем выбирает категорию, и выбирает объект. Выбрав объект пользователю предоставляется информация по объекту. |
| Тип варианта | Основной |

|  |  |
| --- | --- |
| Название варианта | **Просмотр данных категории, в которых нет объектов.** |
| Цель | Получение сведений по объектам звездного неба |
| Действующие лица | Пользователь |
| Краткое описание | Пользователь выбирает режим «Энциклопеди», затем выбирает категорию, и выбирает объект. Выбрав категорию пользователю предоставляется информация по категории. |
| Тип варианта | Основной |

|  |  |
| --- | --- |
| Название варианта | **Игровой процесс** |
| Цель | Прохождение игры |
| Действующие лица | Пользователь |
| Краткое описание | Пользователь выбирает режим «Игра», затем нажимает на кнопку «Начать игру», выбирает тему игры, и выбирает ответ на вопрос из представленных вариантов. |
| Тип варианта | Основной |

|  |  |
| --- | --- |
| Название варианта | **Добавление статистики** |
| Цель | Добавить статистики после прохождения игры пользователем. |
| Действующие лица | Система |
| Краткое описание | После прохождение игры пользователь попадает на экран результатов. Программа сохраняет результат в файл |
| Тип варианта | Дополнительный |

|  |  |
| --- | --- |
| Название варианта | **Просмотр статистики** |
| Цель | Посмотреть общую статистику по играм |
| Действующие лица | Пользователь |
| Краткое описание | Пользователь выбирает режим «Игра», нажимает на кнопку «Статистика». На окне появляется отображается статистика |
| Тип варианта | Основной |

|  |  |
| --- | --- |
| Название варианта | **Удаление статистики** |
| Цель | Удалить статистику |
| Действующие лица | Пользователь |
| Краткое описание | Пользователь выбирает режим «Игра», нажимает на кнопку «Статистика», и нажимает на кнопку «Удалить статистику», в появившемся диалоговом окне пользователь нажимает на кнопку «Да». Статистика обнуляться и записывается в файл |
| Тип варианта | Дополнительный |

Приведем варианты использования «Просмотр данных в энциклопедии в категория, имеющих объекты» и «Игровой процесс» подробно, таблица N:

Вариант использования «**Просмотр данных об объектах**»

|  |  |
| --- | --- |
| Действия пользователя | Отклик системы |
| 1. Пользователь выбирает режим «Энциклопедия» | 1. Система отрисовывает активити выбора категорий |
| 1. Пользователь выбирает категорию | 1. Система отрисовывает активити со списком объектов, относящихся к этой категории. |
| 1. Пользователь выбирает объект | 1. Система отрисовывает активити с информацией по этому объекту. |

Вариант использование «**Игровой процесс**»

|  |  |
| --- | --- |
| Действия пользователя | Отклик системы |
| 1. Пользователь выбирает режим «Игра» | 1. Система отрисовывает активити с меню игры. |
| 1. Пользователь нажимает на кнопку «Начать игру» | 1. Система отрисовывает активити с выбором тем игры |
| 1. Пользователь выбирает тему | 1. Система случайным образом выбирает 10 вопросов и отрисовывает активити с вопросами |
| 1. Пользователь отвечает поочерёдно на 10 вопросов | 1. Система отрисоввывает активити с результатов ответов и сохраняет результат в файл. |

Диаграмма вариантов использования (рисунок N) позволяет наглядно представить ожидаемое поведение системы и уточнить варианты использования данной программы.



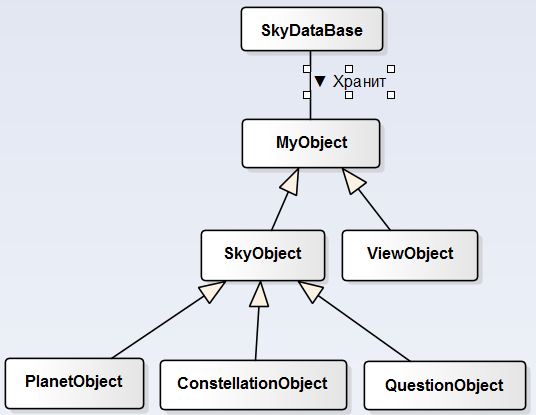
Рисунок 1 - Диаграмма вариантов использования

Действующее лицо пользователь – внешняя к программному обеспечению сущность, взаимодействующая с ним с целью получения или предоставления какой-либо информации.

Полученная диаграмма вариантов использования отражает типичное взаимодействие пользователя с разрабатываемой программой и показывает процедуры, реализующие его конкретные задачи.

**1.4 Построение концептуальный модели предметной области**

Структурная схема на рисунке N отражает состав и взаимодействие основных объектов системы. Структурная схема представленная в виде объектной декомпозиции.



На рисунке видно, что в базе хранятся все объекты предметной области.

**1.5 Описание поведения. Системные события**

Анализирую вариант использования «**Просмотр данных в энциклопедии в категории, имеющей объекты**», определяем, что пользователь должен инициализировать 3 события. Покажем эти события на диаграмме N.



Проанализировав вариант использования «**Игровой процесс**» выделяем 4 события, которые пользователь должен инициализировать.



На основе двух диаграмм последовательностей системы получает список системных операций:



Опишем некоторые из системных операций:

|  |  |
| --- | --- |
| Раздел | Описание |
| Имя | ***Выбрать режим()*** |
| Обязанности | *Отправить пользователя на активити, соответствующую выбранному режиму.* |
| Тип | *Системная* |
| Ссылки | *Вариант использования Просмотр данных об объектах* |
| Примечания | *—* |
| Исключения | *—* |
| Вывод | *—* |
| Предусловия | *Предполагается на выбор два режима* |
| Постусловия | *Пользователь попадает на активити, выбранного режима* |

|  |  |
| --- | --- |
| Раздел | Описание |
| Имя | ***Выбор тему()*** |
| Обязанности | *Отправить пользователя на активити с вопросами по теме.* |
| Тип | *Системная* |
| Ссылки | *Вариант использования Игровой процесс* |
| Примечания | *Предусмотреть выборку из базы данных 10 случайных вопросов по теме* |
| Исключения | 1. *Если в базе данных по выбранной теме нет 10 вопросов, то вывести все имеющиеся* 2. *Если в базе дынных нет вопросов по выбранной теме, то сообщить пользователю об этом* |
| Вывод | *—???????* |
| Предусловия | *Предлагается выбрать тему для игры* |
| Постусловия | *Пользователь попадает на активити, где ему надо ответить на 10 вопросов* |

|  |  |
| --- | --- |
| Раздел | Описание |
| Имя | ***Выбор ответ()*** |
| Обязанности | *Получить ответ польщователя* |
| Тип | *Системная* |
| Ссылки | *Вариант использования Игровой процесс* |
| Примечания | *Предусмотреть смену вопроса после выбора ответа* |
| Исключения | *—* |
| Вывод | *—* |
| Предусловия | *Пользователю предлагается вопрос и 4 варианта ответа* |
| Постусловия | *Смена вопроса на следующий* |

**1.6 Построение диаграмм деятельностей**

Построим диаграммы деятельностей, уточняющие варианты использования «Просмотр данных об объектах» и «Игровой процесс».





**2. Проектирование программного обеспечения**

**2.1 Разработка структуры программного обеспечения**

Анализ концептуальной модели и вариантов использования позволяет выделить следующие группы классов:

* Пользовательский интерфейс – классы, реализующие объекты интерфейса с пользователем
* Библиотека интерфейсных компонентов – классы, реализующие интерфейсные компоненты кнопки, текстовые поля и т.п.
* Объекты управления – классы, реализующие объекты в предметной области системы
* Объекты базы данных – классы, реализующие объекты предметной области.
* Интерфейсы базы данных – классы, реализующие интерфейс с базой данных
* База данных
* Базовые структуры данных – классы, реализующие внутренние структуры данных, такие как списки, деревья, множества.
* Обработка ошибок – классы исключений, реализующие обработку нештатных ситуаций



**2.2 Определение отношений между объектами**

Список классов-кандидатов, полученный на основе анализа концептуальной модели предметной области и варианта использования Просмотр данных об объектах.

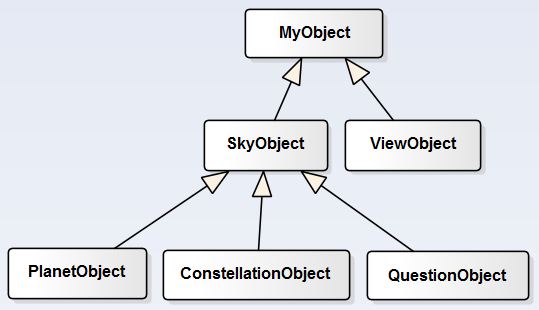


Диаграмма классов пакета Объекты базы

Список некоторых классов-кандидатов пакета Пользовательский интерфейс

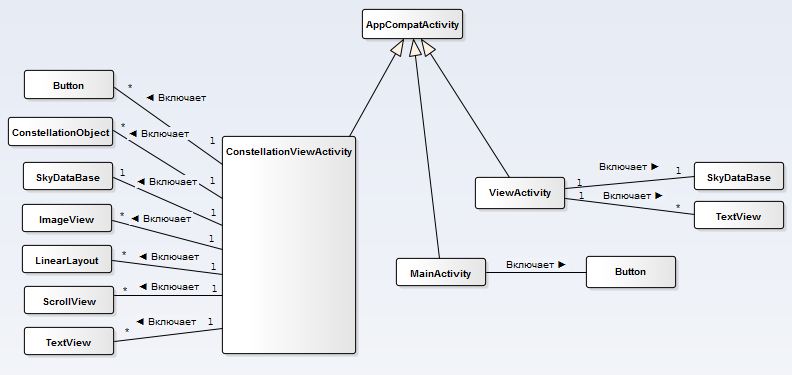


Диаграмма классов пакета Пользовательский интерфейс

**2.3 Диаграммы последовательностей**

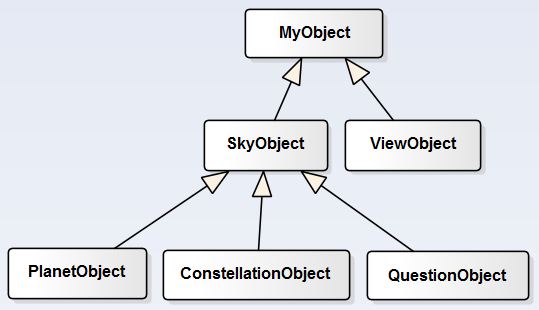
Разработать диаграмму последовательностей для сценария Отображение объектов(фрагмент варианта использования Просмотр данных в энциклопедии)

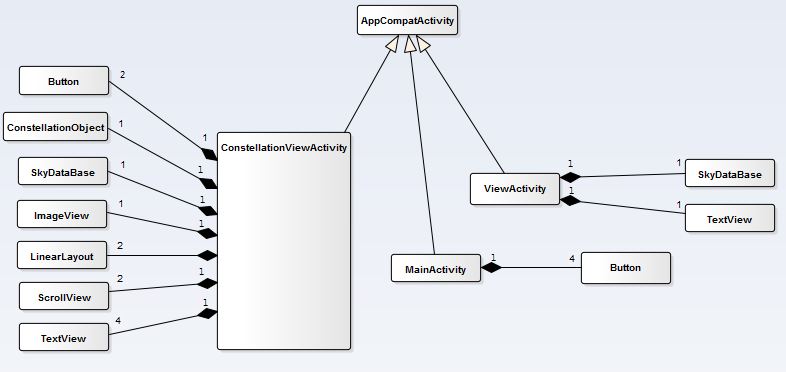


И диаграмма последовательностей для сценария Ответы на вопрос



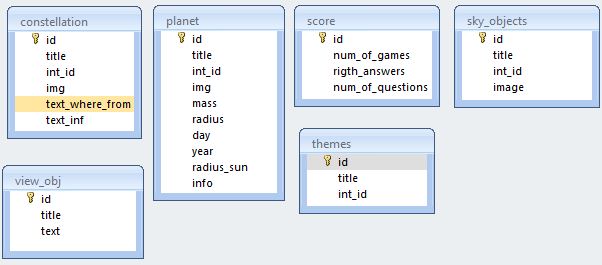
**2.4 Уточнение отношений классов**





**2** **Проектирование структуры и компонентов программного продукта.**

**2.1 Проектирование структуры базы данных**



В базе создано 6 таблиц:

* constellation
* planet
* score
* sky\_objects
* view\_obj
* themes

В таблице constellation хранится информация по созвездиям:

|  |  |
| --- | --- |
| Id | Идентификатор |
| Title | Название созвездия |
| Int\_id | Уникальный идентификатор в приложении |
| Img | Название картинки |
| Text\_where\_from | Информация, где можно найти созвездие на небе |
| Text\_inf | Дополнительная информация |

В таблице planet хранится информация по планетам:

|  |  |
| --- | --- |
| id | Идентификатор |
| Title | Название планеты |
| Int\_id | Уникальный идентификатор в приложении |
| Img | Название картинки |
| Mass | Масса планеты |
| Radius | Средний радиус планеты |
| Day | Продолжительность дня |
| Year | Продолжительность года |
| Radius\_sun | Радиус орбиты |
| info | Дополнительная информация |

Таблица score хранит информация о статистике:

|  |  |
| --- | --- |
| Id | Идентификатор |
| Num\_of\_games | Количество проведённых игр |
| Right\_answers | Количество правильных ответов |
| Num\_of\_questions | Количество вопросов |

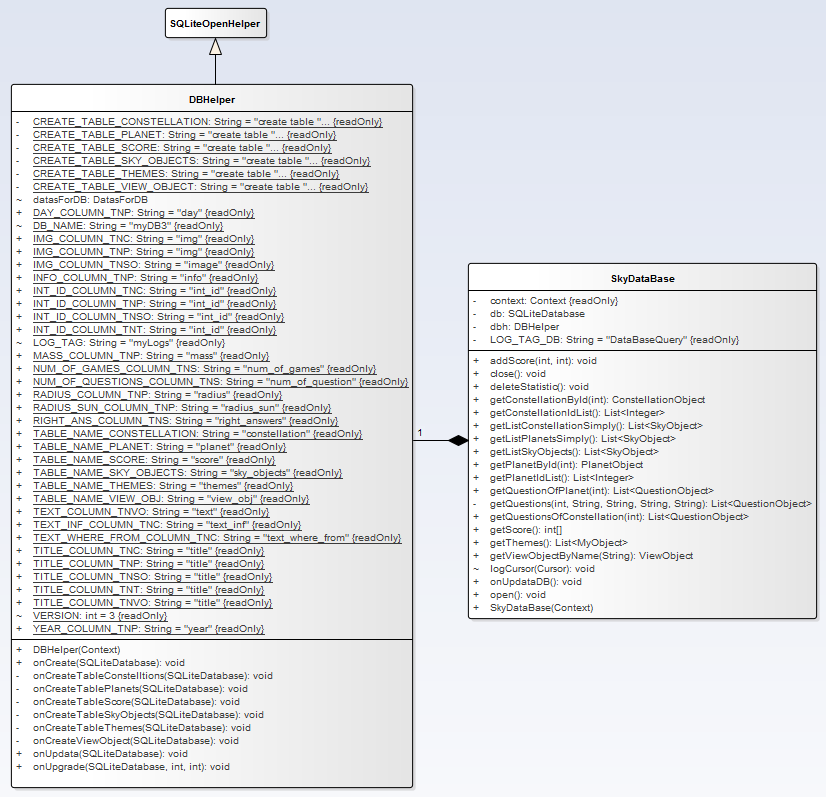
В таблице sky\_objects хранятся объекты звездного неба:

|  |  |
| --- | --- |
| Id | Идентификатор |
| Title | Название |
| Int\_id | Уникальный идентификатор в приложении |
| Img | Название картинки |

В таблице view\_obj хранится информация по космическим объектам:

|  |  |
| --- | --- |
| Id | Идентификатор |
| Title | Название |
| Text | Информация по объектам |

**2.2 Проектирование классов реализующий работу с базой данных**

****

Класс DBHelper содержит все необходимую информацию о структуре база дынных и имеет следующие методы:

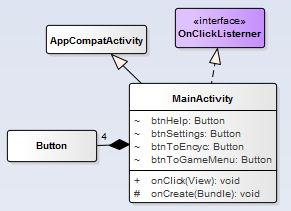
|  |  |
| --- | --- |
| DBHelper() | Конструктор |
| onCreate() | Создание структуры БД |
| onUpgrade() | Изменение существующей структуры БД |
| onUpdata() | Обновление БД |
| onCreateTableThemes() | Создание таблицы theme |
| onCreateTableScore() | Создание таблицы score |
| onCreateTableSkyObjects() | Создание таблицы sky\_objects |
| onCreateTableConstelltions() | Создание таблицы constelltion |
| onCreateTablePlanets() | Создание таблицы planet |
| onCreateViewObject() | Создание таблицы view\_objects |

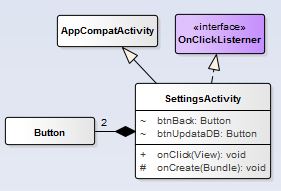
Класс SkyDataBase содерзит следующие методы:

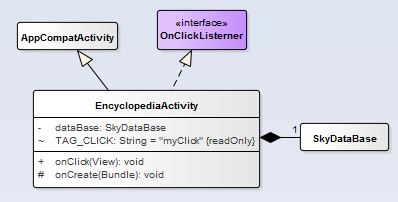
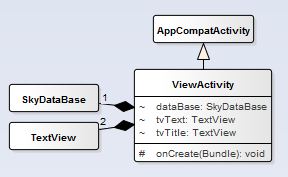
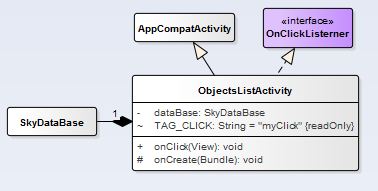
|  |  |
| --- | --- |
| SkyDataBase() | Конструктор класса |
| onUpdataDB() | Обновление БД |
| getListSkyObjects() | Получение списка объектов неба |
| getListConstellationSimply() | Получение упрощенного списка созвездий |
| getConstellationById() | Получение созвездия по имени |
| getConstellationIdList() | Получение списка id созвездий |
| getListPlanetsSimply() | Получение упрощенного списка планет |
| getPlanetById() | Получение планеты по id |
| getPlanetIdList() | Получение списка id планет |
| getThemes() | Получение списка тем |
| getQuestions() | Получение списка вопросов |
| getQuestionsOfConstellation() | Получение списка вопросов созвездий |
| getQuestionOfPlanet() | Получение списка вопросов планет |
| addScore() | Добавление статистики |
| getScore() | Получение статистики |
| deleteStatistic() | Удаление статистики |
| getViewObjectByName() | Получение списка объектов звездного неба |
| logCursor() | Вывод результатов запросов в лог-файл |

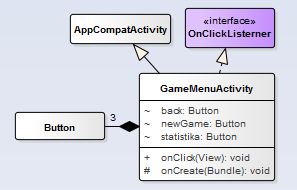
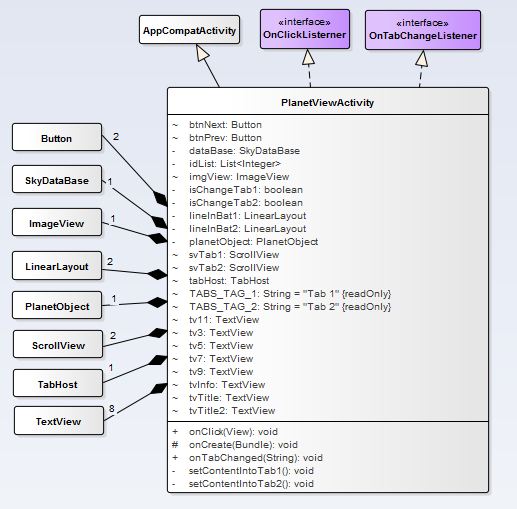
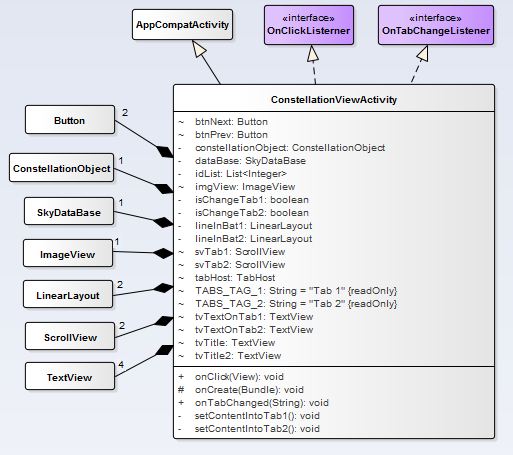
**2.3 Проектирование интерфейса программного продукта**

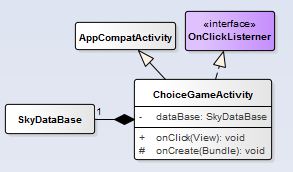
**2.3.1 Проектирование диаграммы классов интерфейсной части**

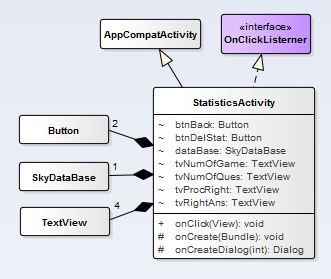
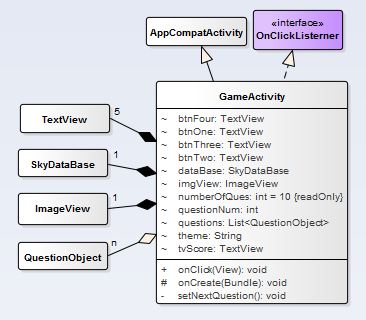
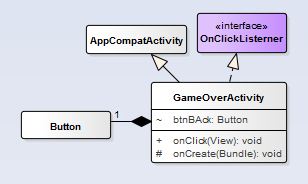












**2.3.2 Проектирование структуры меню.**

Разработанная структурная схема меню, детализирующая пользовательских интерфейс, представленная на рисунке N.

## 

## 2.3.3 Построение графа состояний интерфейса

При проектировании системы были определены основные состояния, а также события, по которым система переходит из одного состояния в другое. На рисунке N представлены графы состояния интерфейса разработанной системы.





