

ПЕТРОЗАВОДСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНСТИТУТ МАТЕМАТИКИ И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ
КАФЕДРА ИНФОРМАТИКИ И МАТЕМАТИЧЕСКОГО ОБЕСПЕЧЕНИЯ

Направление подготовки бакалавриата

09.03.04 - Программная инженерия

Отчет по дисциплине

МЕТОДЫ ТЕСТИРОВАНИЯ ПО

Выполнил:

студент 4 курса группы 22407

В. В. Марков

Преподаватели:

К. А. Кулаков, лектор

М. А. Крышень, руководитель практики

Содержание

1	Объект тестирования	4
1.1	Описание объекта тестирования	4
1.2	Функциональные требования к объекту тестирования	4
2	Стратегия тестирования	4
2.1	Архитектура приложения	4
2.2	Описание модулей	5
2.3	Стратегия блочного тестирования	9
2.4	Стратегия интеграционного тестирования	10
2.5	Стратегия аттестационного тестирования	10
2.6	Стратегия специального тестирования	11
3	Описание тестов	11
3.1	Блочное тестирование	11
3.1.1	Тест Б-1	11
3.1.2	Тест Б-2	12
3.1.3	Тест Б-3	12
3.1.4	Тест Б-4	12
3.1.5	Тест Б-5	13
3.1.6	Тест Б-6	13
3.2	Интеграционное тестирование	13
3.2.1	Тест И-1	13
3.2.2	Тест И-2	14
3.2.3	Тест И-3	15
3.2.4	Тест И-4	15
3.2.5	Тест И-5	16
3.2.6	Тест И-6	16
3.2.7	Тест И-7	17
3.2.8	Тест И-8	18
3.2.9	Тест И-9	18
3.2.10	Тест И-10	19
3.2.11	Тест И-11	19
3.2.12	Тест И-12	20

3.2.13	Тест И-13	21
3.3	Аттестационное тестирование	22
3.3.1	Тест А-1	22
3.3.2	Тест А-2	22
3.3.3	Тест А-3	22
3.3.4	Тест А-4	23
3.3.5	Тест А-5	23
3.3.6	Тест А-6	23
3.4	Специальное тестирование	24
3.4.1	Тест С-1	24
3.4.2	Тест С-2	25
3.4.3	Тест С-3	25
3.4.4	Тест С-4	25
3.4.5	Тест С-5	26
3.4.6	Тест С-6	26
3.5	Пример реализации тестирования	26
4	Результаты тестирования	27
4.1	Отчет о проведении тестирования	27
4.2	Журнал найденных ошибок	30
4.3	Покрытие кода	30
4.4	Результат	31

1 Объект тестирования

1.1 Описание объекта тестирования

Объектом тестирования является мобильное приложение "PetrSU Mobile" для ОС Android. Приложение предоставляет пользователю возможность просмотра ленты новостей ПетрГУ и поиска расписания учебной группы. Данные расписания получаются путем взаимодействия со сторонним API для расписания, данные новостей получаются путем web-скрейпинга ленты новостей формата RSS с сайта ПетрГУ. Приложение реализовано на языке Java.

1.2 Функциональные требования к объекту тестирования

- Просмотр списка новостей
- Отображение подробного описания выбранной новости
- Просмотр расписания выбранной группы
- Выбор предпочитаемой группы для получения расписания

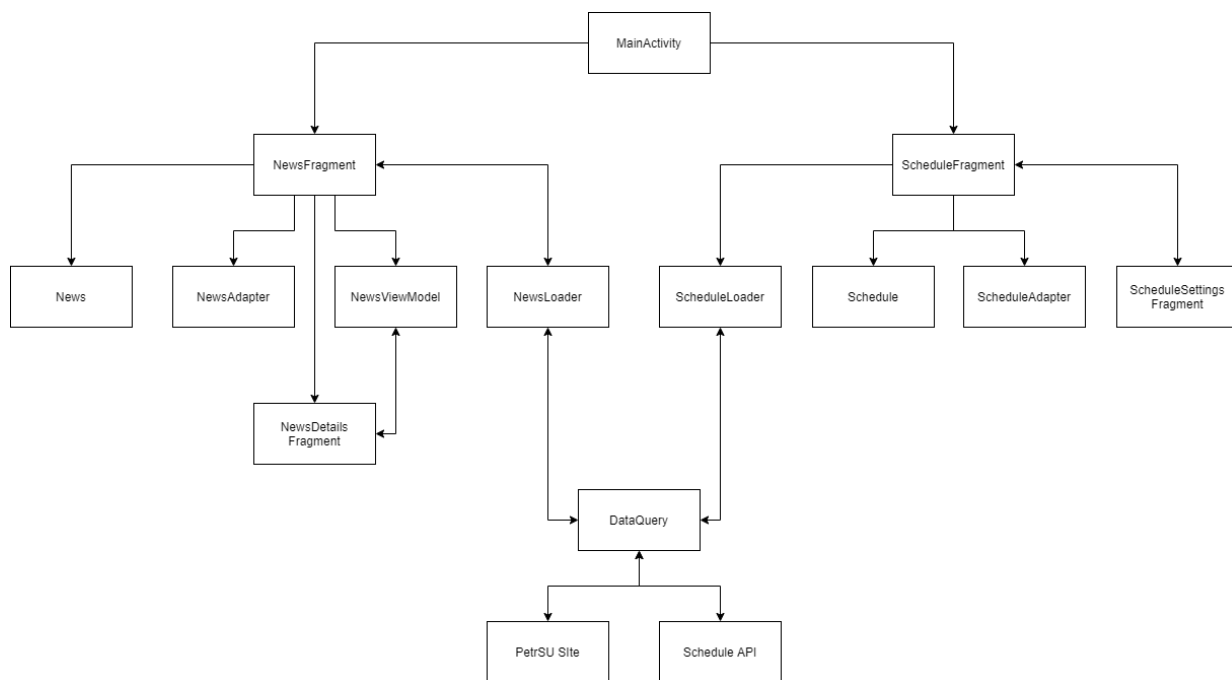
2 Стратегия тестирования

2.1 Архитектура приложения

Модули:

- MainActivity - отвечает за навигацию между фрагментами
- NewsFragment - отвечает за отображение списка новостей
- NewsDetailsFragment - отвечает за отображение подробной информации о выбранной ячейке новости

- ScheduleFragment - отвечает за отображение списка расписания выбранной группы
- ScheduleSettingsFragment - отвечает за редактирование и сохранение номера группы
- DataQuery - отвечает за запросы и обработку получаемых данных



2.2 Описание модулей

- News

Назначение: модель объекта Новость.

Элемент проверочного списка News:

```

public class News {
    private String title = "Доценты Института иностранных языков
    ПетрГУ приняли участие в первом онлайн-салоне для
    выпускников программы Фулбрайта";
    private String description = "Онлайн-салон прошел на
    платформе Zoom и был посвящен теме 'Year 2020 in Remote'";
    private String link = "https://petsu.ru/news/2020/92508/dotsenty-instituta-i";
    private Date date = "Fri, 25 Dec 2020 12:03:00 +0300";
    private String imageUrl = "http://petsu.ru/files/news_notice_event/2020/12/24
  
```

```
        /thumbnails/300_1608816116_ikgudfrqnm.jpg";  
    }  
}
```

Функции:

- public News() - конструктор класса
- public <Type> get<Name>() - Набор геттеров класса

- NewsFragment

Назначение: отображение списка новостей

Функции:

- void initLoaderManager(int loaderId) - получает уникальный идентификатор для менеджера загрузки и инициализирует его для получения списка новостей

- NewsDetailsFragment

Назначение: отображение подробной информации выбранной ячейки новости

Функции:

- void displayDetails(News news) - получает объект Новости для демонстрации подробной информации на экране

- Schedule

Назначение: модель объекта Schedule. В тестировании не участвует

Элемент проверочного списка Schedule:

```
public class Schedule {  
    private String classroom = "";  
    private String startTime = "09:00";  
    private String endTime = "10:25";  
    private String lecturer = "Жуков Артем Владимирович";  
    private String title = "Управление ИТ-сервисами и контентом";  
    private String type = "Лабораторные занятия ДИСТАНТ Дистант";  
}
```

```
    private Integer number = 1;
}
```

Функции:

- public Schedule() - конструктор класса
- public <Type> get<Name>() - Набор геттеров класса

● ScheduleWeek

Назначение: модель объекта ScheduleWeek.

Элемент проверочного списка ScheduleWeek:

```
public class ScheduleWeek {
    private ArrayList<Schedule> scheduleArrayList;
    private String nameWeek = "Понедельник";
}
```

Функции:

- public ScheduleWeekType() - конструктор класса
- public <Type> get<Name>() - Набор геттеров класса

● ScheduleWeekType

Назначение: модель объекта ScheduleWeekType.

Элемент проверочного списка ScheduleWeekType:

```
public class ScheduleWeekType {
    private ArrayList<ScheduleWeek> scheduleWeekArrayList;
    private String weekType = "denominator";
}
```

Функции:

- public ScheduleWeekType() - конструктор класса
- public <Type> get<Name>() - Набор геттеров класса

- ScheduleFragment

Функции:

- void initLoaderManager(int loaderId) - получает уникальный идентификатор для менеджера загрузки и инициализирует его для получения списка расписания

- ScheduleSettingsFramgent

Назначение: выбор группы для получения расписания Функции:

- void bindPreferenceSummaryToValue(Preference preference) - сохраняет введенное значение расписания в системе

- DataQuery

Назначение: получение данных из сети интернет

Функции:

- ArrayList<News> fetchNewsData() - при вызове отправляет запрос на сайт ПетрГУ для получения XML-списка новостей
- ArrayList<News> parseNewsData(NodeList nodeList) - получает XML-список данных и преобразует его в список новостей

Пример правильного формата XML-документа

```
<item>
    <title>Title</title>
    <description>Lorem ipsum</description>
    <link>url-link</link>
    <pubDate>Wed, 16 Dec 2020 15:16:00 +0300</pubDate>
    <enclosure url="urlPhoto" type="image/jpeg"/>
</item>
```

- ArrayList<ScheduleWeekType> fetchScheduleData(String urlString) -

получает ссылку для запроса данных API для получения JSON-объекта расписания группы

- `ArrayList<ScheduleWeekType> parseScheduleData(NodeList nodeList)`
- получает JSON-объект данных и преобразует его в список расписания

Пример правильного формата JSON-объекта

```
{  
    number: 1,  
    title: "Управление ИТ-сервисами и контентом",  
    start-time: "09:00",  
    end-time: "10:25",  
    lecturer: "Жуков Артем Владимирович",  
    type: "Лабораторные занятия ДИСТАНТ Дистант",  
    classroom: "",  
    group: 22407  
}
```

- MainActivity

Назначение: навигация между экранами приложения

Функции:

- `void loadFragment(Fragment fragment)` - получает фрагмент для его демонстрации на экране

2.3 Стратегия блочного тестирования

Тестированию подлежат следующие функции:

- `parseNewsData`
- `parseScheduleData`

Тестирование будут проводиться путем вызова функций и сравнения результата функции с тестовыми данными

2.4 Стратегия интеграционного тестирования

Для проведения интеграционного тестирования необходимо определить все возможные входные данные, соответствующие им ожидаемые результаты. Интеграционное тестирование будет проходить с помощью Android-фреймворка Espresso.

Тестированию подлежат следующие взаимодействия модулей:

- MainActivity -> NewsFragment/ScheduleFragment
- NewsFragment -> DataQuery
- NewsFragment -> NewsDetailsFragment
- ScheduleFragment -> DataQuery
- ScheduleFragment -> ScheduleSettingsFragment
- ScheduleSettingsFragment -> DataQuery

2.5 Стратегия аттестационного тестирования

В ходе аттестационного тестирования будет протестирована работоспособность приложения и его возможность осуществлять заявленный функционал. Аттестационные тесты покрывают ранее перечисленные функциональные требования. Аттестационное тестирование происходит автоматическим способом путем использования фреймворка Espresso.

2.6 Стратегия специального тестирования

Приложение будет протестировано на устройствах, имеющих различную версию операционной системы и разрешение экрана. Само тестирование происходит в формате ручной проверки

Версии ОС:

- Android 8.0
- Android 9.0
- Android 10.0

Разрешения экрана:

- 1440x2880
- 1080x1920
- 480x800

3 Описание тестов

3.1 Блочное тестирование

3.1.1 Тест Б-1

Модуль: DataQuery

Функция: parseNewsData

Тип: позитивный

Описание: преобразование XML-списка данных в список объектов News

Входящие данные: XML-список данных правильного формата(указан в 2.2), проверочный список News(указан в 2.2)

Ожидаемый результат: возвращение списка объектов News

3.1.2 Тест Б-2

Модуль: DataQuery

Функция: parseNewsData

Тип: негативный

Описание: преобразование XML-списска данных в список объектов News

Входящие данные: XML-список неправильного формата, пустой список News

Ожидаемый результат: возвращение пустого списка элементов News

3.1.3 Тест Б-3

Модуль: DataQuery

Функция: parseNewsData

Тип: негативный

Описание: преобразование XML-списска данных в список объектов News

Входящие данные: пустой XML-список, пустой список News

Ожидаемый результат: возвращение пустого списка элементов News

3.1.4 Тест Б-4

Модуль: DataQuery

Функция: parseScheduleData

Тип: позитивный

Описание: преобразование JSON данных в список объектов ScheduleWeekType

Входящие данные: JSON-Объект расписания правильного формата(указан в 2.2), проверочный список Schedule(указан в 2.2)

Ожидаемый результат:возвращение списка объектов ScheduleWeekType

3.1.5 Тест Б-5

Модуль: DataQuery

Функция: parseScheduleData

Тип: негативный

Описание: преобразование JSON данных в список объектов ScheduleWeekType

Входящие данные: JSON-Объект расписания неправильного, пустой список Schedule

Ожидаемый результат:возвращение пустого списка объектов ScheduleWeekType

3.1.6 Тест Б-6

Модуль: DataQuery

Функция: parseScheduleData

Тип: негативный

Описание: преобразование JSON данных в список объектов ScheduleWeekType

Входящие данные: пустой JSON-Объект, пустой список Schedule

Ожидаемый результат:возвращение пустого списка объектов ScheduleWeekType

3.2 Интеграционное тестирование

3.2.1 Тест И-1

Описание: Проверка отображения и сравнение выводимых данных на экране NewsDetails и хранимых в списке объектов News

Ход тестирования:

- Обращение к функции DataQuery().fetchNewsData
- Отправка запроса на получение данных с сайта ПетрГУ
- Получение данных

- Парсинг данных в функции parseNewsData
- Возращение данных

Тип: Положительный

Модули: DataQuery(fetchNewsData, parseNewsData)

Входящие данные: Проверочный элемент News

Косвенные данные: Нет

Ожидаемый результат: Полученные данные получены и имеют правильную структуру

3.2.2 Тест И-2

Описание: Проверка получения списка новостей

Ход тестирования:

- Обращение к функции DataQuery().fetchNewsData
- Отправка запроса на получение данных с сайта ПетрГУ
- Получение данных
- Парсинг данных в функции parseNewsData
- Возращение данных

Тип: Положительный

Модули: DataQuery(fetchNewsData, parseNewsData)

Входящие данные: Проверочный список элементов News

Косвенные данные: Нет

Ожидаемый результат: Полученные данные получены и имеют правильную структуру

3.2.3 Тест И-3

Описание: Проверка получения списка новостей

Ход тестирования:

- Обращение к функции `DataQuery().fetchNewsData`
- Отправка запроса на получение данных с сайта ПетрГУ
- Получение данных
- Парсинг данных в функции `parseNewsData`
- Возращение данных

Тип: Отрицательный

Модули: `DataQuery(fetchNewsData, parseNewsData)`

Входящие данные: Отсутствие интернет-соединения

Косвенные данные: Нет

Ожидаемый результат: Ошибка получения данных, возвращение пустого списка

3.2.4 Тест И-4

Описание: Проверка получения списка новостей

Ход тестирования:

- Обращение к функции `DataQuery().fetchNewsData`
- Отправка запроса на получение данных с сайта ПетрГУ
- Получение данных
- Парсинг данных в функции `parseNewsData`
- Возращение данных

Тип: Отрицательный

Модули: `DataQuery(fetchNewsData, parseNewsData)`

Входящие данные: Формат полученных данных неверен или пуст

Косвенные данные: Нет

Ожидаемый результат: Ошибка обработки данных, возвращение пустого списка

3.2.5 Тест И-5

Описание: Проверка отображения списка новостей

Ход тестирования:

- Нажатие на кнопку экрана Новости
- Обращение к функции `DataQuery().fetchNewsData`
- Получение данных в `NewsFragment`
- Отправка данных в `NewsAdapter`
- Сравнение хранимых и выводимых в ячейках данных

Тип: Положительный

Модули: `NewsFragment -> NewsAdapter`

Входящие данные: Проверочный список элементов `News`

Косвенные данные: Нет

Ожидаемый результат: выводимые данные выбранной ячейки совпадают с хранимыми

3.2.6 Тест И-6

Описание: Проверка отображения списка новостей

Ход тестирования:

- Нажатие на кнопку экрана Новости
- Обращение к функции `DataQuery().fetchData`
- Получение данных в `NewsFragment`
- Отправка данных в `NewsAdapter`
- Сравнение хранимых и выводимых в ячейках данных

Тип: Отрицательный

Модули: `NewsFragment` -> `NewsAdapter`

Входящие данные: Пустой список данных

Косвенные данные: Нет

Ожидаемый результат: Пустой экран

3.2.7 Тест И-7

Описание: Проверка отображения списка расписания

Ход тестирования:

- Нажатие на кнопку экрана `Schedule`
- Обращение к функции `DataQuery().fetchSchedule`
- Получение данных в `ScheduleFragment`
- Отправка данных в `ScheduleAdapter`
- Сравнение хранимых и выводимых в ячейках данных

Тип: Положительный

Модули: `ScheduleFragment` -> `ScheduleAdapter`

Входящие данные: Проверочный список элементов `Schedule`

Косвенные данные: Нет

Ожидаемый результат: выводимые данные выбранной ячейки совпадают с хранимыми

3.2.8 Тест И-8

Описание: Проверка отображения списка расписания

Ход тестирования:

- Нажатие на кнопку экрана Schedule
- Обращение к функции `DataQuery().fetchSchedule`
- Получение данных в `ScheduleFragment`
- Отправка данных в `ScheduleAdapter`
- Сравнение хранимых и выводимых в ячейках данных

Тип: Отрицательный

Модули: `ScheduleFragment` -> `ScheduleAdapter`

Входящие данные: Пустой список данных

Косвенные данные: Нет

Ожидаемый результат: Пустой экран

3.2.9 Тест И-9

Описание: Проверка получения списка расписания

Ход тестирования:

- Обращение к функции `DataQuery().fetchScheduleData`
- Отправка запроса на получение данных с сайта ПетрГУ
- Получение данных
- Парсинг данных в функции `parseScheduleData`

- Возвращение данных

Тип: Положительный

Модули: DataQuery(fetchScheduleData, parseScheduleData)

Входящие данные: Проверочный список элементов News

Косвенные данные: Нет

Ожидаемый результат: Полученные данные получены и имеют правильную структуру

3.2.10 Тест И-10

Описание: Проверка получения списка расписания

Ход тестирования:

- Обращение к функции DataQuery().fetchScheduleData
- Отправка запроса на получение данных с сайта ПетрГУ
- Получение данных
- Парсинг данных в функции parseScheduleData
- Возвращение данных

Тип: Отрицательный

Модули: DataQuery(fetchScheduleData, parseScheduleData)

Входящие данные: Отсутствие интернет-соединения

Косвенные данные: Нет

Ожидаемый результат: Ошибка получения данных, возвращение пустого списка

3.2.11 Тест И-11

Описание: Проверка получения списка расписания

Ход тестирования:

- Обращение к функции `DataQuery().fetchScheduleData`
- Отправка запроса на получение данных с сайта ПетрГУ
- Получение данных
- Парсинг данных в функции `parseScheduleData`
- Возвращение данных

Тип: Отрицательный

Модули: `DataQuery(fetchScheduleData, parseScheduleData)`

Входящие данные: Формат полученных данных неверен или пуст

Косвенные данные: Нет

Ожидаемый результат: Ошибка обработки данных, возвращение пустого списка

3.2.12 Тест И-12

Описание: Проверка изменения расписания через настройки

Ход тестирования:

- Нажатие на кнопку экрана `Schedule`
- Нажатие на кнопку экрана `ScheduleSettingsFragment`
- Нажатие на строку с номером группы
- Ввод номера группы
- Возвращение на экран `ScheduleFragment`
- Обращение к функции `DataQuery().fetchSchedule`
- Получение данных в `ScheduleFragment`
- Отправка данных в `ScheduleAdapter`

- Сравнение хранимых и выводимых в ячейках данных

Тип: Положительный

Модули: ScheduleFragment, ScheduleSettingsFragment, DataQuery

Входящие данные: Номер существующей группы

Косвенные данные: Нет

Ожидаемый результат: Расписание изменилось, выводимые данные совпадают с хранимыми

3.2.13 Тест И-13

Описание: Проверка изменения расписания через настройки

Ход тестирования:

- Нажатие на кнопку экрана Schedule
- Нажатие на кнопку экрана ScheduleSettingsFragment
- Нажатие на строку с номером группы
- Ввод номера группы
- Возвращение на экран ScheduleFragment
- Обращение к функции DataQuery().fetchSchedule
- Получение данных в ScheduleFragment
- Отправка данных в ScheduleAdapter
- Сравнение хранимых и выводимых в ячейках данных

Тип: Отрицательный

Модули: ScheduleFragment, ScheduleSettingsFragment, DataQuery

Входящие данные: Номер несуществующей группы

Косвенные данные: Нет

Ожидаемый результат: Расписание изменилось, экран пуст

3.3 Аттестационное тестирование

3.3.1 Тест А-1

Описание: Приложение запускается

Тип: Общий

Ход тестирования:

- Открытие приложения

Ожидаемый результат: Приложение запустилось, открылся экран новостей

3.3.2 Тест А-2

Описание: Проверка навигации между экранами

Тип: Общий

Ход тестирования:

- Нажатие на кнопку экрана News
- Нажатие на кнопку экрана Schedule

Ожидаемый результат: экраны последовательно корректно открылись

3.3.3 Тест А-3

Описание: Выбор ячейки новости. Открытие экрана подробного описания

Тип: Общий

Ход тестирования:

- Нажатие на кнопку экрана News
- Нажатие на ячейку новости

Ожидаемый результат: открылся экран NewsDetails

3.3.4 Тест А-4

Описание: Проверка отображения расписания

Тип: Общий

Ход тестирования:

- Нажатие на кнопку экрана Schedule

Ожидаемый результат: открылся экран Schedule, на экране отображается список с расписанием

3.3.5 Тест А-5

Описание: Проверка изменения расписания

Тип: Общий

Ход тестирования:

- Нажатие на кнопку экрана Schedule
- Нажатие на кнопку экрана Settings
- Нажатие на строку с номером группы
- Ввод существующего номера группы
- Возвращение на предыдущий экран

Ожидаемый результат: открылся экран Schedule, на экране отображается список с новым расписанием

3.3.6 Тест А-6

Описание: Проверка сохранения состояния номера группы в настройках

Тип: Общий

Ход тестирования:

- Нажатие на кнопку экрана Schedule
- Нажатие на кнопку экрана Settings
- Нажатие на строку с номером группы
- Ввод номера группы
- Закрытие приложения
- Открытие приложения
- Повторение пунктов 1-2
- Возвращение на экран ScheduleFragment
- Сравнение хранимых и выводимых в ячейках данных

Ожидаемый результат: Номер группы в настройках сохранился, данные выбранной группы совпадают с выводимыми

3.4 Специальное тестирование

3.4.1 Тест С-1

Описание: Проверка корректности отображения и взаимодействия с интерфейсом в приложении.

Тип: Общий

Входящие данные: Версия ОС 8.0

Косвенные данные: Нет

Ожидаемый результат: Приложение работает исправно и выглядит согласно заданным гайдлайнам графического интерфейса

3.4.2 Тест С-2

Описание: Проверка корректности отображения и взаимодействия с интерфейсом в приложении.

Тип: Общий

Входящие данные: Версия ОС 9.0

Косвенные данные: Нет

Ожидаемый результат: Приложение работает исправно и выглядит согласно заданным гайдлайнам графического интерфейса

3.4.3 Тест С-3

Описание: Проверка корректности отображения и взаимодействия с интерфейсом в приложении.

Тип: Общий

Входящие данные: Версия ОС 10.0

Косвенные данные: Нет

Ожидаемый результат: Приложение работает исправно и выглядит согласно заданным гайдлайнам графического интерфейса

3.4.4 Тест С-4

Описание: Проверка корректности отображения и взаимодействия с интерфейсом в приложении.

Тип: Общий

Входящие данные: Разрешение 1440x2880

Косвенные данные: Нет

Ожидаемый результат: Приложение работает исправно и выглядит согласно заданным гайдлайнам графического интерфейса

3.4.5 Тест С-5

Описание: Проверка корректности отображения и взаимодействия с интерфейсом в приложении.

Тип: Общий

Входящие данные: Разрешение 1080x1920

Косвенные данные: Нет

Ожидаемый результат: Приложение работает исправно и выглядит согласно заданным гайдлайнам графического интерфейса

3.4.6 Тест С-6

Описание: Проверка корректности отображения и взаимодействия с интерфейсом в приложении.

Тип: Общий

Входящие данные: Разрешение 480x800

Косвенные данные: Нет

Ожидаемый результат: Приложение работает исправно и выглядит согласно заданным гайдлайнам графического интерфейса

3.5 Пример реализации тестирования

В качестве примера рассмотрим реализацию блочного теста Б-4.

```

@Before
public void createTestModel(){
    schedule = new Schedule( classroom: "",
        startTime: "09:00",
        endTime: "10:25",
        lecturer: "Жуков Артем Владимирович",
        title: "Управление ИТ-сервисами и контентом",
        type: "Лабораторные занятия ДИСТАНТ Дистант",
        number: 1);
}

@Test
public void getScheduleList() {
    ArrayList<ScheduleWeekType> schedules = DataQuery.parseScheduleData(object);
    Schedule schedule1 = schedules.get(0).getScheduleWeekArrayList().get(0).getScheduleArrayList().get(0);
    assertEquals(schedule.getType(), schedule1.getType());
    assertEquals(schedule.getClassroom(), schedule1.getClassroom());
    assertEquals(schedule.getEndTime(), schedule1.getEndTime());
    assertEquals(schedule.getStartTime(), schedule1.getStartTime());
    assertEquals(schedule.getLecturer(), schedule1.getLecturer());
    assertEquals(schedule.getNumber(), schedule1.getNumber());
    assertEquals(schedule.getTitle(), schedule1.getTitle());
}

```

До начала теста создается проверочный элемент Schedule и тестовый JSON-объект. После начала происходит вызов функции parseScheduleData и выбор 1 элемента списка из всего расписания. Далее происходит проверка всех параметров полученного и тестового классов.

4 Результаты тестирования

4.1 Отчет о проведении тестирования

Тест	Дата	Тестирующий	Попытки	Результат
Б-1	07.12.2020	Марков В.В.	1	Пройден
Б-2	07.12.2020	Марков В.В.	1	Пройден
Б-3	07.12.2020	Марков В.В.	1	Пройден
Б-4	07.12.2020	Марков В.В.	1	Пройден
Б-5	07.12.2020	Марков В.В.	1	Пройден
Б-6	07.12.2020	Марков В.В.	1	Пройден

Тест	Дата	Тестирующий	Попытки	Результат
И-1	07.12.2020	Марков В.В.	1	Пройден
И-2	07.12.2020	Марков В.В.	1	Пройден
И-3	07.12.2020	Марков В.В.	1	Пройден
И-4	07.12.2020	Марков В.В.	1	Пройден
И-5	07.12.2020	Марков В.В.	1	Пройден
И-6	07.12.2020	Марков В.В.	1	Пройден
И-7	07.12.2020	Марков В.В.	1	Пройден
И-8	07.12.2020	Марков В.В.	1	Пройден
И-9	07.12.2020	Марков В.В.	1	Пройден
И-10	07.12.2020	Марков В.В.	1	Пройден
И-11	07.12.2020	Марков В.В.	1	Пройден
И-12	07.12.2020	Марков В.В.	1	Пройден
И-13	07.12.2020	Марков В.В.	1	Пройден

Тест	Дата	Тестирующий	Попытки	Результат
А-1	07.12.2020	Марков В.В.	1	Пройден
А-2	07.12.2020	Марков В.В.	1	Пройден
А-3	07.12.2020	Марков В.В.	1	Пройден
А-4	07.12.2020	Марков В.В.	1	Пройден
А-5	07.12.2020	Марков В.В.	1	Пройден
А-6	07.12.2020	Марков В.В.	1	Пройден

Тест	Дата	Тестирующий	Попытки	Результат
С-1	07.12.2020	Марков В.В.	2	Пройден
С-2	07.12.2020	Марков В.В.	2	Пройден
С-3	07.12.2020	Марков В.В.	2	Не пройден
С-4	07.12.2020	Марков В.В.	2	Не пройден
С-5	07.12.2020	Марков В.В.	2	Пройден
С-6	07.12.2020	Марков В.В.	2	Не пройден

4.2 Журнал найденных ошибок

Номер	Дата	Тест	Ожидаемый результат	Фактический результат
1	07.12.2020	C-3	Приложение работает исправно и выглядит согласно заданным гайдлайнам графического интерфейса	На экране Новости не отображаются картинки новостей
2	07.12.2020	C-4	Приложение работает исправно и выглядит согласно заданным гайдлайнам графического интерфейса	Приложение работает исправно. Расстояние между элементами не соответствуют гайдлайнам
3	07.12.2020	C-6	Приложение работает исправно и выглядит согласно заданным гайдлайнам графического интерфейса	Расстояние между элементами не соответствуют гайдлайнам, часть функций стали недоступны из-за наложения объектов

4.3 Покрытие кода

Для генерации отчета по покрытию кода тестами использовался фреймворк Ясосо

Результат покрытия: 78%

4.4 Результат

В ходе тестирования было выявлено 3 ошибки. Найденные ошибки были незначительны и устранены после обнаружения. Разработанное приложение удовлетворяет всем функциональным требованиям.