Санкт-Петербургский политехнический университет Петра Великого

Кафедра компьютерных систем и программных технологий

**Отчёт по лабораторной работе №4**

**Дисциплина**: Проектирование мобильных приложений

**Тема**: RecyclerView

Выполнил студент гр. 3530901/70203 И.Д. Иванов

(подпись)

Преподаватель И.С. Егорова

(подпись)

“\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_\_ 2019 г.

Санкт-Петербург

2019

**1. Цели** **работы**

* Ознакомиться с принципами работы adapter-based views;
* Получить практические навыки разработки адаптеров для view.

**2. Программа работы**

**2.1. Задача 1.** Знакомство с библиотекой (unit test)

Ознакомьтесь со strict mode библиотеки, проиллюстрировав его работу unit-тестом.

В файле BibDatabaseTest.java содержатся тесты для проверки создания библиотеки (setup()), чтения одной записи (getFirstEntry()), а также normal (не strict) mode (normalModeDoesNotThrowException()).

Предложено написать два теста для демонстрации работы флагов BibConfig#strict и BibConfig#shuffle.

С установленным флагом BibConfig#strict работает искусственное ограничение: в памяти нельзя хранить более BibConfig#maxValid=20 записей одновременно. При извлечении maxValid+1й записи 1-я извлеченная запись становится невалидной (при доступе к полям кидаются исключения). Тестирование заключалось в том, чтобы убедиться в наличии исключения при извлечении maxValid+1й записи.

С флагом BibConfig#shuffle записи, находящиеся в определенном порядке в исходном файле, перемешиваются в случайном порядке для хранения внутри библиотеки. Тестирование заключалось в том, чтобы убедиться в отличии записей, находящихся на одинаковой позиции, при разных значениях флага.

Код успешно проходящих тестов представлен в приложении 1.

**2.2. Задача 2.** Знакомство с RecyclerView

Напишите Android приложение, которое выводит все записи из bibtex файла на экран, используя предложенную библиотеку и RecyclerView.

**2.2.1 Однородный список** (задача обычной сложности). Все записи имеют одинаковый формат (@article).

Написано приложение, в котором создается библиотека и на экран выводятся все ее записи. Создан класс BibAdapter, унаследованный от RecyclerView.Adapter. Adapter – класс, который отвечает за связь элементов java-кода с view-компонентами. В BibAdapter реализовано 3 метода:

-onCreateViewHolder(parent: ViewGroup, viewType: Int): BibViewHolder –вызывается для создания объекта типа ViewHolder, в конструктор которого необходимо передать созданный view-компонент, с которым будут связываться java объекты. В этом методе указан отдельно созданный layout-ресурс – «внешний вид» элемента списка.

-onBindViewHolder(holder: BibViewHolder, position: Int) – отвечает за связь java объекта и view. Здесь происходит запись данный во view-компоненты.

-getItemCount() – сообщает количество элементов в списке.

В MainActivity происходит создание библиотеки и запись данных в RecycleView – передача созданных экземпляров классов BibAdapter и LayoutManager. LayoutManager отвечает за способ расположения элементов на экране – линейный, в виде сетки, в виде шахматной сетки. В данном случае используется LinearLayoutManager, который обеспечивает линейное отображение.

Также, для удобства представления информации создан data-класс BibElem.

Выводится следующая информация о записи: заголовок статьи, авторы, название журнала, в котором опубликована статья, объём статьи, а также год публикации.

Запущенное приложение показано на рис.1.

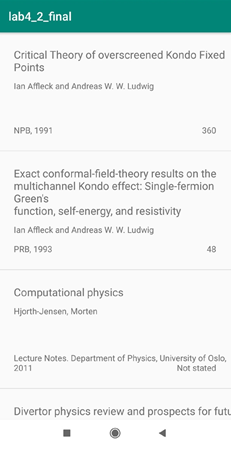


Рис.1.

Код MainActivity, BibElem и BibAdapter представлен в приложении 2.

**2.3. Задача 3.** Бесконечный список

Сделайте список из предыдущей задачи бесконечным: после последнего элемента все записи повторяются, начиная с первой.

Данная задача была решена в классе BibAdapter:

Метод getItemCount() теперь возвращает Int.MAX\_VALUE вместо количества записей в исходном файле. В методе onBindViewHolder индекс новой записи считается по модулю количества записей в библиотеке (это обеспечивает повтор записей).

Код изменённого BibAdapter представлен в приложении 3.

**3. Выводы**

В ходе выполнения лабораторной работы были изучены принципы работы adapter-based views, а также получены навыки работы с RecyclerView.

**Приложение 1**

@Test(expected = IllegalStateException.class)  
public void strictModeThrowsException() {  
 BibConfig cfg = database.getCfg();  
 cfg.strict = true;  
  
 BibEntry first = database.getEntry(0);  
 for (int i = 0; i < cfg.maxValid; i++) {  
 BibEntry unused = database.getEntry(0);  
 }  
 first.getType();//should throw IllegalStateException  
}  
  
@Test  
public void shuffleFlag() throws IOException {  
 BibConfig cfg = database.getCfg();  
 cfg.shuffle=false;  
  
 try (InputStreamReader reader = new InputStreamReader(getClass().getResourceAsStream("/referencesForShuffle.bib"))) {  
 database = new BibDatabase(reader,cfg);  
 }  
 Types first = database.getEntry(0).getType();  
 cfg.shuffle=true;  
 boolean check=false;  
  
 for(int i=0; i<10;i++){  
 try (InputStreamReader reader = new InputStreamReader(getClass().getResourceAsStream("/referencesForShuffle.bib"))) {  
 database = new BibDatabase(reader,cfg);  
 }  
 if (first != database.getEntry(0).getType()) {  
 check = true;  
 break;  
 }  
 }  
  
 Assert.*assertTrue*(check);  
 cfg.shuffle=false;  
 try (InputStreamReader reader = new InputStreamReader(getClass().getResourceAsStream("/referencesForShuffle.bib"))) {  
 database = new BibDatabase(reader,cfg);  
 }  
 Assert.*assertEquals*(first, database.getEntry(0).getType());  
}

**Приложение 2**

**MainActivity:**

package com.example.lab4\_2\_final  
  
import androidx.appcompat.app.AppCompatActivity  
import android.os.Bundle  
import androidx.recyclerview.widget.DividerItemDecoration  
import androidx.recyclerview.widget.LinearLayoutManager  
import kotlinx.android.synthetic.main.activity\_main.\*  
import name.ank.lab4.BibDatabase  
import name.ank.lab4.Keys  
  
class MainActivity : AppCompatActivity() {  
  
 private lateinit var bibAdepter: BibAdapter  
  
  
 private val bibData = *mutableListOf*<BibElem>()  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.*activity\_main*)  
  
 val articles = *resources*.openRawResource(R.raw.*articles*)  
  
 val database = BibDatabase(articles.*reader*())  
 val size = database.*size* for (i in 0 *until* size) {  
 val entry = database.getEntry(i)  
 bibData.add(  
 BibElem(  
 entry.getField(Keys.AUTHOR)?:"Not stated", entry.getField(Keys.TITLE)?:"Not stated",  
 entry.getField(Keys.JOURNAL)?:"Not stated", entry.getField(Keys.YEAR)?:"Not stated",  
 entry.getField(Keys.VOLUME)?:"Not stated"  
 )  
 )  
  
 }  
  
 val recyclerLayoutManager = LinearLayoutManager(this)  
 bibAdepter = BibAdapter(bibData)  
 val dividerItemDecoration = DividerItemDecoration(  
 my\_recycler\_view.*context*,  
 recyclerLayoutManager.*orientation* )  
 my\_recycler\_view.*apply* **{** *adapter* = bibAdepter  
 setHasFixedSize(true)  
 *layoutManager* = recyclerLayoutManager  
 addItemDecoration(dividerItemDecoration)  
 **}** }  
}

**BibElem:**

package com.example.lab4\_2\_final  
  
data class BibElem(  
 val author: String,  
 val title: String,  
 val journal: String,  
 val year: String,  
 val volume:String  
)

**BibAdapter:**

package com.example.lab4\_2\_final  
  
import android.view.LayoutInflater  
import android.view.View  
import android.view.ViewGroup  
import androidx.recyclerview.widget.RecyclerView  
import kotlinx.android.synthetic.main.bib\_elem.view.\*  
  
class BibAdapter(private val bibData : List<BibElem>) :  
 RecyclerView.Adapter<BibAdapter.BibViewHolder>() {  
  
 inner class BibViewHolder(val view: View) : RecyclerView.ViewHolder(view)  
  
 override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): BibViewHolder =  
 BibViewHolder(LayoutInflater.from(parent.*context*).inflate(R.layout.*bib\_elem*, parent,false))  
  
 override fun onBindViewHolder(holder: BibViewHolder, position: Int) {  
 holder.view.*apply* **{** title\_text\_view.*text* = bibData[position].title  
 author\_text\_view.*text* = bibData[position].author  
 jourlan\_text\_view.*text* = "%s, %s".*format*(bibData[position].journal,bibData[position].year)  
 volume\_text\_view.*text* = bibData[position].volume  
 **}** }  
  
 override fun getItemCount() = bibData.size  
}

**Приложение 3**

**BibAdapter:**

package com.example.lab4\_3  
  
import android.view.LayoutInflater  
import android.view.View  
import android.view.ViewGroup  
import androidx.recyclerview.widget.RecyclerView  
import kotlinx.android.synthetic.main.bib\_elem.view.\*  
  
class BibAdapter(private val bibData : List<BibElem>) :  
 RecyclerView.Adapter<BibAdapter.BibViewHolder>() {  
  
 inner class BibViewHolder(val view: View) : RecyclerView.ViewHolder(view)  
  
 override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): BibViewHolder =  
 BibViewHolder(LayoutInflater.from(parent.*context*).inflate(R.layout.*bib\_elem*, parent,false))  
  
 override fun onBindViewHolder(holder: BibViewHolder, position: Int) {  
 holder.view.*apply* **{** title\_text\_view.*text* = bibData[position % bibData.size].title  
 author\_text\_view.*text* = bibData[position % bibData.size].author  
 jourlan\_text\_view.*text* = "%s, %s".*format*( bibData[position % bibData.size].journal, bibData[position % bibData.size].year)  
 volume\_text\_view.*text* = bibData[position % bibData.size].volume  
 **}** }  
  
 override fun getItemCount() = Int.MAX\_VALUE  
}