Департамент образования Воронежской области

Государственное бюджетное профессиональное образовательное учреждение

Воронежской области «Борисоглебский техникум промышленных

и информационных технологий»

ОТЧЕТ

по учебной практике

УП.05.01 Проектирование и разработка информационных систем

|  |  |
| --- | --- |
| Выполнил студент \_\_3\_\_ курса  спец. 09.02.07 «Информационные системы и программирование»  Группа 3.2ИСИП\_\_  Юшкин.А.А  (ФИО)  Дата сдачи \_\_31.05.2023 г.\_\_\_\_ | Проверил  преподаватель  Пеньков А.С.  (ФИО)  Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

Борисоглебск

2022 –2023 учебный год

Оглавление

[1. ВЫБОР СРЕДСТВ РАЗРАБОТКИ ПРОЕКТА 3](#_Toc135825015)

[4](#_Toc135825016)

[ТЕМА 2. РАЗРАБОТКА ТЕХНИЧЕСКОГО ЗАДАНИЯ 5](#_Toc135825017)

[ТЕМА 3. СИСТЕМА КОНТРОЛЯ ВЕРСИЙ 7](#_Toc135825018)

[ТЕМА 4. СБОР ДАННЫХ ДЛЯ СОЗДАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ 9](#_Toc135825019)

[ТЕМА 5. СОЗДАНИЕ МАКЕТА ИНФОРМАЦИОННОЙ СИСТЕМЫ. 11](#_Toc135825022)

[ТЕМА 7. MATERIAL DESIGN 15](#_Toc135825029)

[ТЕМА 8. РАБОТА СО СПИСОЧНЫМИ ЭЛЕМАНТАМИ ИНТЕРФЕЙСА 18](#_Toc135825030)

[ТЕМА 9. ОБЪЕКТО-ОРИЕНТИРОВАННЫЙ ПОДХОД В РАЗРАБОТКЕ КОДА ИНФОРМАЦИОННОЙ СИСТЕМЫ 23](#_Toc135825031)

[ТЕМА 10. СОЗДАНИЕ ПРОЦЕДУР ОБРАБОТКИ ИНФОРМАЦИИ 33](#_Toc135825032)

[ТЕМА 11. ОБЕСПЕЧЕНИЕ ДОПОЛНИТЕЛЬНОЙ ФУНКЦИОНАЛЬНОЙ СИСТЕМЫ 39](#_Toc135825033)

[ТЕМА 12. INTENTS 42](#_Toc135825034)

[ТЕМА 13. ТЕСТИРОВАНИЕ ПРИЛОЖЕНИЯ 45](#_Toc135825035)

[ТЕМА 14. ОТЛАДКА ПРИЛОЖЕНИЯ 48](#_Toc135825036)

[ТЕМА 15. ЗАЩИТА ПРИЛОЖЕНИЯ.ОЦЕНКА КАЧЕСТВА И НАДЕЖНОСТИ ПРИЛОЖЕНИЯ. 50](#_Toc135825037)

[ТЕМА 16. СОПРОЖДЕНИЕ ПРОГРАММ 52](#_Toc135825038)

[ТЕМА 17. СОПРОВОЖДЕНИЕ ПРОГРАММ 55](#_Toc135825039)

[ТЕМА 18. ИТОГОВОЕ ЗАНЯТИЕ 58](#_Toc135825040)

# ВЫБОР СРЕДСТВ РАЗРАБОТКИ ПРОЕКТА

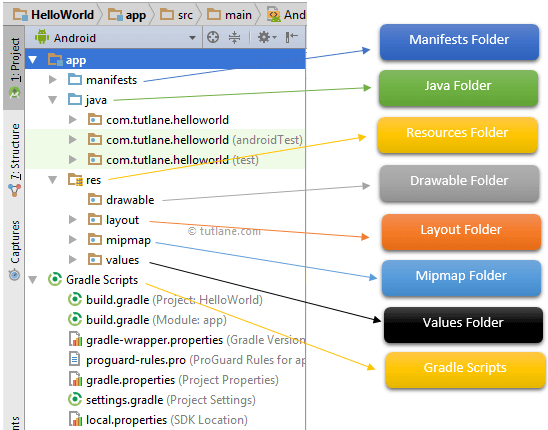
1.Определение ОС Android

Открытая ОС для мобильных телефонов, смартфонов, коммуникаторов, планшетных компьютеров, электронных книг, цифровых проигрывателей, наручных часов, нетбуков и смартбуков, основанная на ядре Linux.

2.Что входит в инструментарий Android разработчика

JDK (Java Development Kit), IDE(Integrated Development Environment), SDK(Android Software Development Kit).

3.Перечислите обязательные элементы в структуре приложения



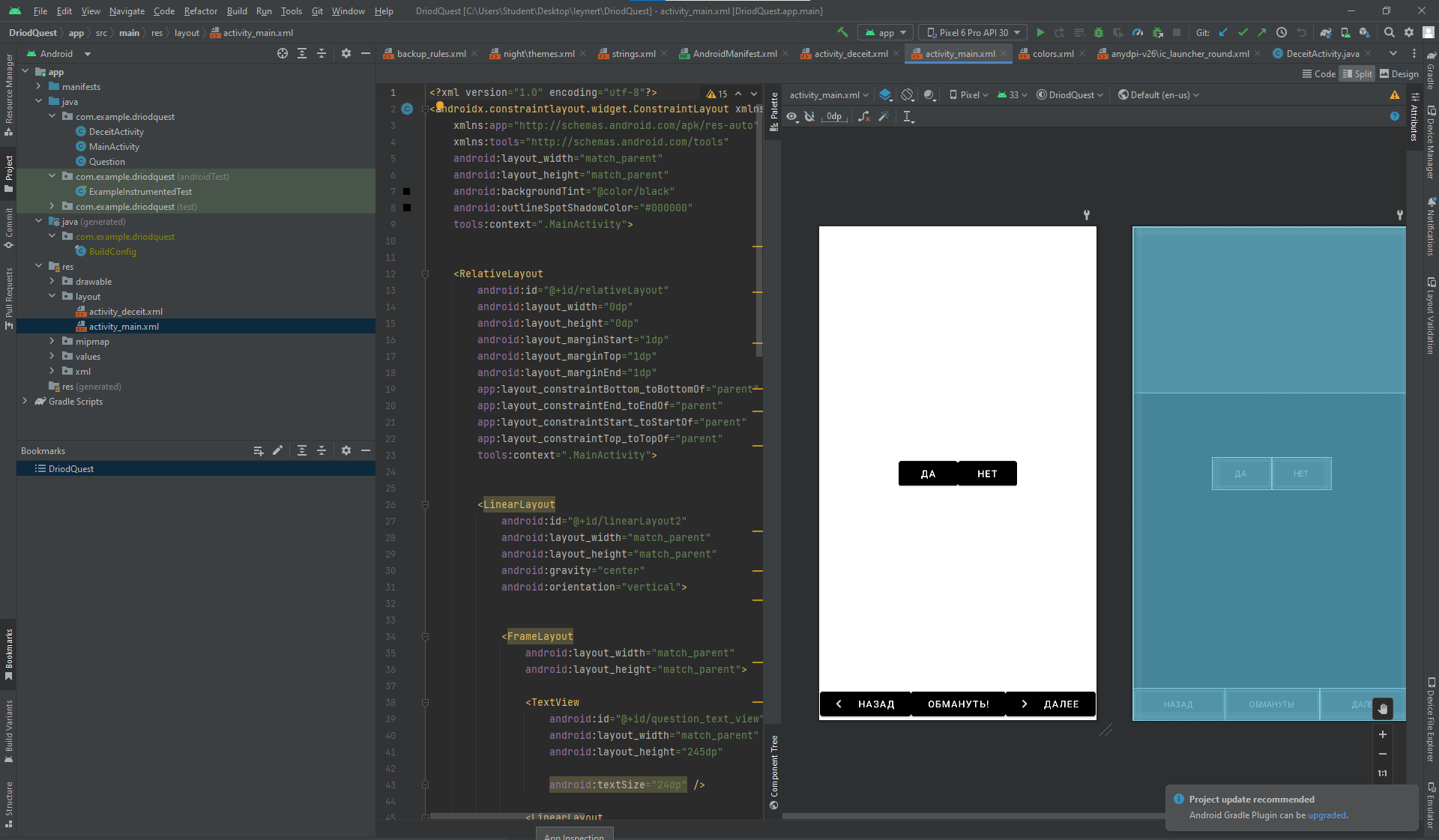
4.Перечислите опциальные элементы в структуре приложения

* src
* gen
* Android Version
* Assets
* libs
* res
  + drawable
  + layout
  + anim
  + values
* AndroidManifest.xml
* default.properties

5. Изобразите жизненный цикл android приложения

# 

# ТЕМА 2. РАЗРАБОТКА ТЕХНИЧЕСКОГО ЗАДАНИЯ



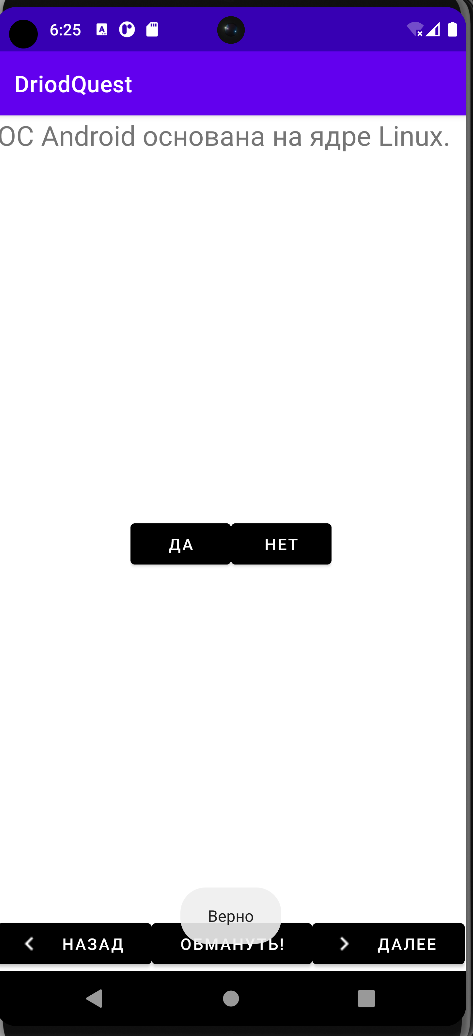
<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:gravity="center"  
 android:orientation="vertical">  
  
  
 <TextView  
 android:id="@+id/textView"  
 android:layout\_width="347dp"  
 android:layout\_height="wrap\_content"  
 android:text="ОС Android основана на ядре Linux"  
 android:textSize="20sp" />  
  
 <LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal">  
  
  
 <Button  
 android:id="@+id/true\_button"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:text="@string/true\_button" />  
  
 <Button  
 android:id="@+id/false\_button"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="20dp"  
 android:layout\_weight="1"  
 android:text="@string/false\_button" />  
 </LinearLayout>  
</LinearLayout>

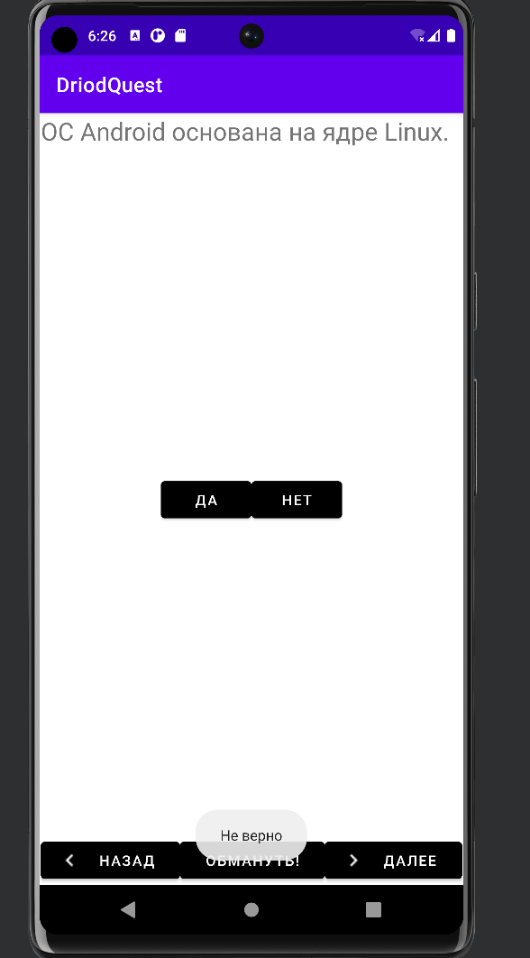
Strings.xml

<resources>  
 <string name="app\_name">DriodQuest</string>  
  
 <string name="true\_button">Да </string>  
 <string name="false\_button">Нет </string>  
 <string name="action\_setting">Setting </string>  
 <string name="correct\_toast">Верно</string>  
 <string name="incorrect\_toast">Не верно</string>

</resources>

}





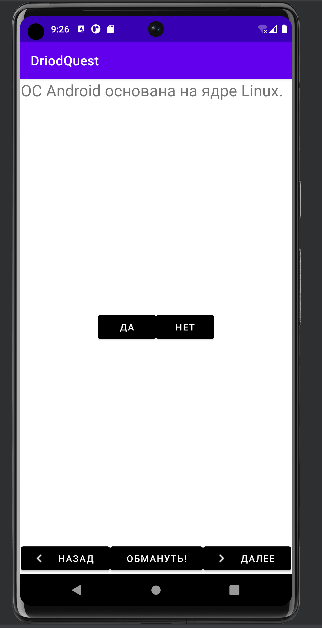
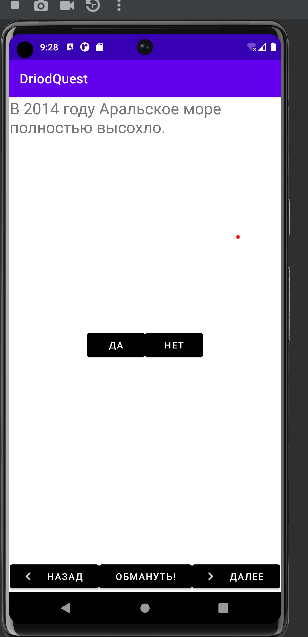
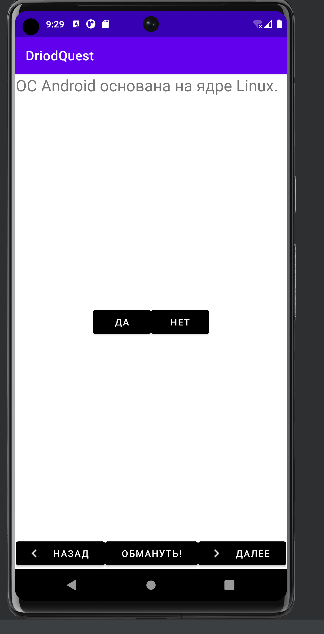
ТЕМА 3. СИСТЕМА КОНТРОЛЯ ВЕРСИЙ

<string name="next\_button">Далее</string>  
<string name="back\_button">Назад</string>

<TextView  
 android:id="@+id/question\_text\_view"  
 android:layout\_width="match\_parent"  
 android:layout\_height="245dp"  
  
 android:textSize="24dp" />

<Button  
 android:id="@+id/back\_button"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:backgroundTint="@color/black"  
 android:contentDescription="@string/back\_button"  
 android:drawableRight="@drawable/baseline\_keyboard\_arrow\_left\_black\_24dp"  
 android:text="@string/back\_button"  
 app:icon="@drawable/baseline\_keyboard\_arrow\_left\_black\_24dp" />

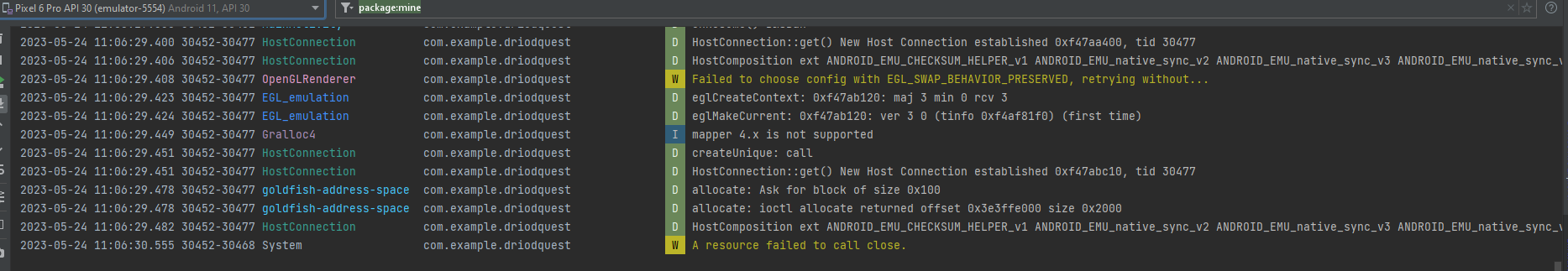
<Button  
 android:id="@+id/next\_button"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:backgroundTint="@color/black"  
 android:contentDescription="@string/next\_button"  
 android:drawableRight="@drawable/baseline\_keyboard\_arrow\_right\_black\_24dp"  
 android:text="@string/next\_button"  
 android:drawablePadding="4dp"  
 android:gravity="center"

# 

# ТЕМА 4. СБОР ДАННЫХ ДЛЯ СОЗДАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ

Log.*d*(*TAG*, "onCreate(Bundle) вызван");

@Override  
public void onStart() {  
 super.onStart();  
 Log.*d*(*TAG*, "onStart() вызван");  
}  
  
@Override  
public void onPause() {  
 super.onPause();  
 Log.*d*(*TAG*, "onPause() вызван");  
}  
  
@Override  
public void onResume() {  
 super.onResume();  
 Log.*d*(*TAG*, "onResume() вызван");  
}  
  
@Override  
public void onStop() {  
 super.onStop();  
 Log.*d*(*TAG*, "onStop() вызван");  
}  
  
@Override  
public void onDestroy() {  
 super.onDestroy();  
 Log.*d*(*TAG*, "onDestroy() вызван");  
}

# Сделала альбомную ориентацию

# 

# ТЕМА 5. СОЗДАНИЕ МАКЕТА ИНФОРМАЦИОННОЙ СИСТЕМЫ.

Создаем кнопуку «Обмануть!»

MainActivity.java

<Button  
 android:id="@+id/deceit\_button"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:backgroundTint="@color/black"  
 android:outlineAmbientShadowColor="@color/black"  
 android:outlineSpotShadowColor="@color/black"  
 android:text="@string/deceit\_button"  
 app:rippleColor="@color/black" />

private Button mDeceitButton;

mDeceitButton = (Button)findViewById(R.id.*deceit\_button*);   
mDeceitButton.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 boolean answerIsTrue = mQuestionBank[mCurrentIndex]  
 .isAnswerTrue();  
 Intent i = DeceitActivity.*newIntent*(MainActivity.this, answerIsTrue);  
 startActivityForResult(i, *REQUEST\_CODE\_DECEIT*);  
 }  
});

boolean answerIsTrue = mQuestionBank[mCurrentIndex]  
 .isAnswerTrue();  
Intent i = DeceitActivity.*newIntent*(MainActivity.this, answerIsTrue);

DeceitActivity.java

private boolean mAnswerIsTrue;  
private TextView mAnswerTextView;  
private Button mShowAnswer;

protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_deceit*);  
 mAnswerIsTrue = getIntent().getBooleanExtra(*EXTRA\_ANSWER\_IS\_TRUE*, false);  
 mAnswerTextView = (TextView) findViewById(R.id.*answer\_text\_view*);  
 mShowAnswer = (Button) findViewById(R.id.*show\_answer\_button*);  
 mShowAnswer.setOnClickListener(new View.OnClickListener() {

Получение результата от дочерней активности

private static final int *REQUEST\_CODE\_DECEIT* = 0;

startActivityForResult(i, *REQUEST\_CODE\_DECEIT*);

Возвращение интента

public static final String *EXTRA\_ANSWER\_SHOWN* = "com.example.my\_practica.answer\_shown";

setAnswerShownResult(true);  
 }  
 });  
  
  
 }  
  
 private void setAnswerShownResult(boolean isAnswerShown) {  
 Intent data = new Intent();  
 data.putExtra(*EXTRA\_ANSWER\_SHOWN*, isAnswerShown);  
 setResult(*RESULT\_OK*, data);  
 }  
}

public static boolean wasAnswerShown(Intent result) {  
 return result.getBooleanExtra(*EXTRA\_ANSWER\_SHOWN*, false);  
}

# Обработка результата

private boolean mIsDeceiter;

@Override  
protected void onActivityResult(int requestCode, int resultCode, Intent data) {  
 super.onActivityResult(requestCode, resultCode, data);  
 if (resultCode != Activity.*RESULT\_OK*) {  
 return;  
 }  
 if (requestCode == *REQUEST\_CODE\_DECEIT*) {  
 if (data == null) {  
 return;  
 }  
  
 mIsDeceiter = DeceitActivity.*wasAnswerShown*(data);  
 }

# Изменение уведомления в зависимости от значения mIsDeceiter (QuestActivity.java)

if (mIsDeceiter) {  
 messageResId = R.string.*judgment\_toast*;  
} else {  
 if (userPressedTrue == answerIsTrue) { messageResId = R.string.*correct\_toast*;  
 } else {  
 messageResId = R.string.*incorrect\_toast*;  
 }

@Override  
public void onClick(View view) {  
 mCurrentIndex = (mCurrentIndex + 1) % mQuestionBank.length;  
 mIsDeceiter = false;  
 updateQuestion();  
  
}

# Результат

# 

# Если мы нажем кнопку нет

# 

# ТЕМА 7. MATERIAL DESIGN

Activity\_main.xml

<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:id="@+id/fragmentContainer"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 >  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical">  
  
 <EditText  
 android:id="@+id/book\_title"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="@string/book\_title\_hint" />  
 </LinearLayout>  
</FrameLayout>

fragment\_book.xml

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical">  
 <EditText  
 android:id="@+id/book\_title"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="@string/book\_title\_hint"  
 />  
</LinearLayout>

strings.xml

<resources>  
 <string name="app\_name">BookDepository</string>  
 <string name="menu\_settings">Settings</string>  
 <string name="title\_activity\_book">BookActivity</string>  
 <string name="book\_title\_hint">Введите название книги.</string>  
 <!-- TODO: Remove or change this placeholder text -->  
 <string name="hello\_blank\_fragment">Hello blank fragment</string>  
  
</resources>

BookFragment2.java

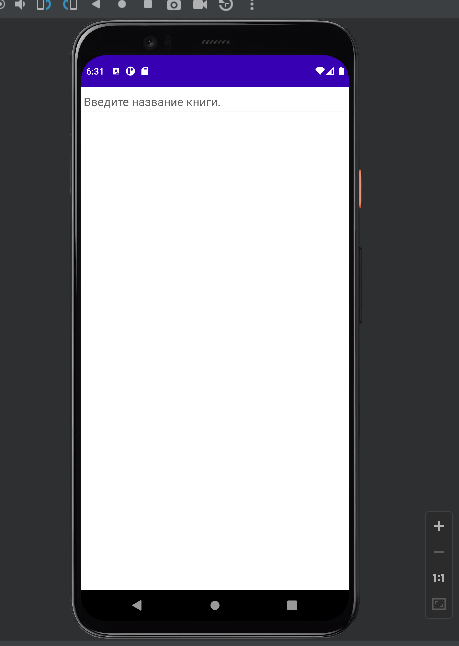
package com.example.bookdepository;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.os.Bundle;  
import android.support.v4.app.Fragment;  
import android.text.Editable;  
import android.text.TextWatcher;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.EditText;  
  
public class BookFragment2 extends Fragment {  
  
 private Book mBook;  
 private EditText mTitleField;  
 @Override  
 public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {  
 View v = inflater.inflate(R.layout.*activity\_main*, container, false);  
 mTitleField = (EditText) v.findViewById(R.id.*book\_title*);  
 mTitleField.addTextChangedListener(new TextWatcher() {  
 @Override  
 public void beforeTextChanged(CharSequence s, int start, int count, int after) {  
  
 }  
  
 @Override  
 public void onTextChanged(CharSequence s, int start, int before, int count) {  
 mBook.setTitle(s.toString());  
 }  
  
 @Override  
 public void afterTextChanged(Editable c) {  
  
 }  
 });  
 return v;  
 }  
  
}

BookActivity.java

package com.example.bookdepository;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.os.Bundle;  
import android.support.v4.app.Fragment;  
import android.support.v4.app.FragmentActivity;  
import android.support.v4.app.FragmentManager;  
  
public class BookActivity extends FragmentActivity {  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 FragmentManager fm = getSupportFragmentManager();  
 Fragment fragment = fm.findFragmentById(R.id.*fragmentContainer*);  
 if (fragment == null) {  
 fragment = new BookFragment2();  
 fm.beginTransaction()  
 .add(R.id.*fragmentContainer*, fragment)  
 .commit();  
 }  
 }  
}

Book.java

package com.example.bookdepository;  
  
import java.util.UUID;  
  
public class Book {  
 private UUID mId; private  
 String mTitle;  
 public Book() {  
 mId = UUID.*randomUUID*(); //Генерирование уникального идентификатора  
 }  
 public UUID getId() {  
 return mId;  
 }  
 public String getTitle() {  
 return mTitle;  
 }  
 public void setTitle(String title) {  
 mTitle = title;  
 }  
}



ТЕМА 8. РАБОТА СО СПИСОЧНЫМИ ЭЛЕМАНТАМИ ИНТЕРФЕЙСА

Book.java

package com.example.bookdepository;  
  
import java.util.Date;  
import java.util.UUID;  
  
public class Book {  
 private UUID mId; private  
 String mTitle;  
 private Date mDate;  
 private boolean mReaded;  
 public Book() {  
 mId = UUID.*randomUUID*(); //Генерирование уникального идентификатора  
 mDate = new Date();  
 }  
  
 public Date getDate() {  
 return mDate;  
 }  
 public void setDate(Date date) {  
 mDate = date;  
 }  
 public boolean isReaded() {  
 return mReaded;  
 }  
  
 public void setReaded(boolean readed) {  
 mReaded = readed;  
 }  
  
 public UUID getId() {  
 return mId;  
 }  
 public String getTitle() {  
 return mTitle;  
 }  
 public void setTitle(String title) {  
 mTitle = title;  
 }  
}

BookActivity.java

package com.example.bookdepository;  
  
import java.util.Date;  
import java.util.UUID;  
  
public class Book {  
 private UUID mId; private  
 String mTitle;  
 private Date mDate;  
 private boolean mReaded;  
 public Book() {  
 mId = UUID.*randomUUID*(); //Генерирование уникального идентификатора  
 mDate = new Date();  
 }  
  
 public Date getDate() {  
 return mDate;  
 }  
 public void setDate(Date date) {  
 mDate = date;  
 }  
 public boolean isReaded() {  
 return mReaded;  
 }  
  
 public void setReaded(boolean readed) {  
 mReaded = readed;  
 }  
  
 public UUID getId() {  
 return mId;  
 }  
 public String getTitle() {  
 return mTitle;  
 }  
 public void setTitle(String title) {  
 mTitle = title;  
 }  
}

BookFragment2.java

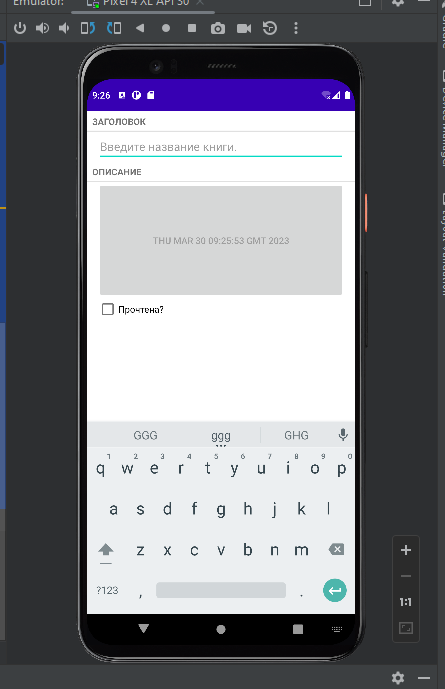
package com.example.bookdepository;  
  
import java.util.Date;  
  
import android.os.Bundle;  
import android.support.v4.app.Fragment;  
import android.text.Editable;  
import android.text.TextWatcher;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.Button;  
import android.widget.CheckBox;  
import android.widget.CompoundButton;  
import android.widget.EditText;  
  
public class BookFragment2 extends Fragment {  
  
 private Book mBook;  
 private EditText mTitleField;  
 private Button mDateButton;  
 private CheckBox mReadedCheckBox;  
 @Override  
 public void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 mBook = new Book();  
 }  
 @Override  
 public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {  
 View v = inflater.inflate(R.layout.*fragment\_book*, container, false);  
 mTitleField = (EditText) v.findViewById(R.id.*book\_title*);  
 mTitleField.addTextChangedListener(new TextWatcher() {  
 @Override  
 public void beforeTextChanged(CharSequence s, int start, int count, int after) {  
  
 }  
  
 @Override  
 public void onTextChanged(CharSequence s, int start, int before, int count) {  
 mBook.setTitle(s.toString());  
 }  
  
 @Override  
 public void afterTextChanged(Editable c) {  
  
 }  
 });  
 mDateButton = (Button) v.findViewById(R.id.*book\_date*);  
 mDateButton.setText(mBook.getDate().toString());  
 mDateButton.setEnabled(false);  
mReadedCheckBox = (CheckBox)v.findViewById(R.id.*book\_readed*);  
mReadedCheckBox.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {  
 @Override  
 public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {  
 mBook.setReaded(isChecked);  
 }  
});  
 return v;  
 }  
  
}

activity\_main.xml

<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:id="@+id/fragmentContainer"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 >  
  
  
</FrameLayout>

fragment\_book.xml

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical">  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="@string/book\_title\_label"  
 style="?android:listSeparatorTextViewStyle"  
 />  
 <EditText  
 android:id="@+id/book\_title"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="16dp"  
 android:layout\_marginRight="16dp"  
 android:hint="@string/book\_title\_hint"  
 />  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="@string/book\_details\_label"  
 style="?android:listSeparatorTextViewStyle"  
 />  
 <Button android:id="@+id/book\_date"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="16dp"  
 android:layout\_marginRight="16dp"  
 android:padding="80dp"  
 />  
 <CheckBox  
 android:id="@+id/book\_readed"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="16dp"  
 android:layout\_marginRight="16dp"  
 android:text="@string/book\_readed\_label"  
 />  
  
</LinearLayout>



ТЕМА 9. ОБЪЕКТО-ОРИЕНТИРОВАННЫЙ ПОДХОД В РАЗРАБОТКЕ КОДА ИНФОРМАЦИОННОЙ СИСТЕМЫ

Book.java

package ru.rsue.leynert.bookdepository;

import java.util.Date;  
import java.util.UUID;  
  
public class Book {  
 private UUID mId; private  
 String mTitle;  
 private Date mDate;  
 private boolean mReaded;  
 public Book() {  
 mId = UUID.*randomUUID*(); //Генерирование уникального идентификатора  
 mDate = new Date();  
 }  
  
 public Date getDate() {  
 return mDate;  
 }  
 public void setDate(Date date) {  
 mDate = date;  
 }  
 public boolean isReaded() {  
 return mReaded;  
 }  
  
 public void setReaded(boolean readed) {  
 mReaded = readed;  
 }  
  
 public UUID getId() {  
 return mId;  
 }  
 public String getTitle() {  
 return mTitle;  
 }  
 public void setTitle(String title) {  
 mTitle = title;  
 }  
}

BookActivity

package ru.rsue.leynert.bookdepository;

import android.os.Bundle;  
import android.support.v4.app.Fragment;  
import android.support.v4.app.FragmentActivity;  
import android.support.v4.app.FragmentManager;  
  
public class BookActivity extends SingleFragmentActivity{  
 @Override  
 protected Fragment createFragment() {  
  
 return new BookFragment2();  
 }  
  
}

BookFragment2

package ru.rsue.leynert.bookdepository;

import java.util.Date;  
  
import android.os.Bundle;  
import android.support.v4.app.Fragment;  
import android.text.Editable;  
import android.text.TextWatcher;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.Button;  
import android.widget.CheckBox;  
import android.widget.CompoundButton;  
import android.widget.EditText;  
  
public class BookFragment2 extends Fragment {  
  
 private Book mBook;  
 private EditText mTitleField;  
 private Button mDateButton;  
 private CheckBox mReadedCheckBox;  
 @Override  
 public void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 mBook = new Book();  
 }  
 @Override  
 public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {  
 View v = inflater.inflate(R.layout.*fragment\_book*, container, false);  
 mTitleField = (EditText) v.findViewById(R.id.*book\_title*);  
 mTitleField.addTextChangedListener(new TextWatcher() {  
 @Override  
 public void beforeTextChanged(CharSequence s, int start, int count, int after) {  
  
 }  
  
 @Override  
 public void onTextChanged(CharSequence s, int start, int before, int count) {  
 mBook.setTitle(s.toString());  
 }  
  
 @Override  
 public void afterTextChanged(Editable c) {  
  
 }  
 });  
 mDateButton = (Button) v.findViewById(R.id.*book\_date*);  
 mDateButton.setText(mBook.getDate().toString());  
 mDateButton.setEnabled(false);  
mReadedCheckBox = (CheckBox)v.findViewById(R.id.*book\_readed*);  
mReadedCheckBox.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {  
 @Override  
 public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {  
 mBook.setReaded(isChecked);  
 }  
});  
 return v;  
 }  
  
}

BookLab.java

package ru.rsue.leynert.bookdepository;

import android.content.Context;  
  
import java.util.ArrayList;  
import java.util.List;  
import java.util.UUID;  
  
public class BookLab {  
 private static BookLab *sBookLab*;  
 private List<Book> mBooks;  
 public static BookLab get(Context context) {  
 if (*sBookLab* == null) {  
 *sBookLab* = new BookLab(context);  
  
 }  
 return *sBookLab*;  
 }  
 private BookLab(Context context) {  
 mBooks = new ArrayList<>();  
 for (int i = 0; i < 100; i++) {  
 Book book = new Book();  
 book.setTitle("Book #" + i);  
 book.setReaded(i % 2 == 0); // Для каждого второго объекта  
 mBooks.add(book);  
 }  
 }  
 public List<Book> getBooks(){  
 return mBooks;  
 }  
 public Book getBook(UUID id) {  
 for (Book book : mBooks) {  
 if (book.getId().equals(id)) {  
 return book;  
 }  
 }  
 return null;  
 }  
}

BookListActivity.java

package ru.rsue.leynert.bookdepository;

import android.support.v4.app.Fragment;  
  
  
public class BookListActivity extends SingleFragmentActivity {  
 @Override  
 protected Fragment createFragment() {  
 return new BookListFragment();  
  
 }  
}

BookListFragment.java

package ru.rsue.leynert.bookdepository;

import android.os.Bundle;  
import android.support.v4.app.Fragment;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.CheckBox;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import androidx.annotation.NonNull;  
import androidx.recyclerview.widget.LinearLayoutManager;  
import androidx.recyclerview.widget.RecyclerView;  
  
import java.util.List;  
  
public class BookListFragment extends Fragment {  
 private RecyclerView mBookRecyclerView;  
 private BookAdapter mAdapter;  
 @Override  
 public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {  
 View view = inflater.inflate(R.layout.*fragment\_book\_list*, container, false);  
 mBookRecyclerView = (RecyclerView) view  
 .findViewById(R.id.*book\_recycler\_view*);  
 mBookRecyclerView.setLayoutManager(new LinearLayoutManager(getActivity()));  
 updateUI();  
 return view;  
 }  
 private void updateUI() {  
 BookLab bookLab = BookLab.*get*(getActivity());  
 List<Book> books = bookLab.getBooks();  
 mAdapter = new BookAdapter(books);  
 mBookRecyclerView.setAdapter(mAdapter);  
 }  
  
 private class BookHolder extends RecyclerView.ViewHolder implements View.OnClickListener {  
 private TextView mTitleTextView;  
 private TextView mDateTextView;  
 private Book mBook;  
 private CheckBox mReadedCheckBox;  
  
 public void bindBook(Book book) { mBook = book;  
 mTitleTextView.setText(mBook.getTitle());  
 mDateTextView.setText(mBook.getDate().toString());  
 mReadedCheckBox.setChecked(mBook.isReaded());  
 }  
 public BookHolder(View itemView) {  
 super(itemView);  
 itemView.setOnClickListener(this);  
 mTitleTextView = (TextView)  
 itemView.findViewById(R.id.*list\_item\_book\_title\_text\_view*); mDateTextView = (TextView)  
 itemView.findViewById(R.id.*list\_item\_book\_date\_text\_view*); mReadedCheckBox = (CheckBox)  
 itemView.findViewById(R.id.*list\_item\_book\_readed\_check\_box*);  
  
 }  
  
 public void onClick(View v) { Toast.*makeText*(getActivity(),  
 mBook.getTitle() + " clicked!", Toast.*LENGTH\_SHORT*)  
 .show();  
 }  
}  
  
  
  
 private class BookAdapter extends RecyclerView.Adapter<BookHolder> {  
 private List<Book> mBooks;  
 public BookAdapter(List<Book> books) {  
 mBooks = books;  
 }  
  
 @Override  
 public BookHolder onCreateViewHolder( ViewGroup parent, int viewType) {  
 LayoutInflater layoutInflater = LayoutInflater.*from*(getActivity());  
 View view = layoutInflater.inflate(R.layout.*list\_item\_book*, parent, false);  
 return new BookHolder(view);  
  
 }  
  
 @Override  
 public void onBindViewHolder(BookHolder holder, int position) {  
 Book book = mBooks.get(position);  
 holder.bindBook(book);  
 }  
  
 @Override  
 public int getItemCount() {  
  
 return mBooks.size();  
 }  
 }  
  
}

SingleFragmentActivity.java

package ru.rsue.leynert.bookdepository;

import android.os.Bundle;  
import android.support.v4.app.Fragment;  
import android.support.v4.app.FragmentActivity;  
import android.support.v4.app.FragmentManager;  
  
public abstract class SingleFragmentActivity extends FragmentActivity {  
 protected abstract Fragment createFragment();  
  
 @Override  
 public void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_fragment*);  
 FragmentManager fm = getSupportFragmentManager();  
 Fragment fragment = fm.findFragmentById(R.id.*fragmentContainer*);  
 if (fragment == null) {  
 fragment = new BookListFragment();//BookFragment2  
 fm.beginTransaction()  
 .add(R.id.*fragmentContainer*, fragment)  
 .commit();  
 }  
 }  
}

activity\_fragment.xml

<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:id="@+id/fragmentContainer"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 >  
  
  
</FrameLayout>

fragment\_book.xml

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:orientation="vertical">  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="@string/book\_title\_label"  
 style="?android:listSeparatorTextViewStyle"  
 />  
 <EditText  
 android:id="@+id/book\_title"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="16dp"  
 android:layout\_marginRight="16dp"  
 android:hint="@string/book\_title\_hint"  
 />  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="@string/book\_details\_label"  
 style="?android:listSeparatorTextViewStyle"  
 />  
 <Button android:id="@+id/book\_date"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="16dp"  
 android:layout\_marginRight="16dp"  
 android:padding="80dp"  
 />  
 <CheckBox  
 android:id="@+id/book\_readed"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="16dp"  
 android:layout\_marginRight="16dp"  
 android:text="@string/book\_readed\_label"  
 />  
  
</LinearLayout>

land\fragment\_book.xml

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical">  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="@string/book\_title\_label"  
 style="?android:listSeparatorTextViewStyle"  
 />  
 <EditText  
 android:id="@+id/book\_title"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="16dp"  
 android:layout\_marginRight="16dp"  
 android:hint="@string/book\_title\_hint"  
 />  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="@string/book\_details\_label"  
 style="?android:listSeparatorTextViewStyle"  
 />  
 <Button android:id="@+id/book\_date"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="16dp"  
 android:layout\_marginRight="16dp"   
 android:padding="80dp"  
 />  
 <CheckBox  
 android:id="@+id/book\_readed"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="16dp"  
 android:layout\_marginRight="16dp"  
 android:text="@string/book\_readed\_label"  
 />  
  
</LinearLayout>

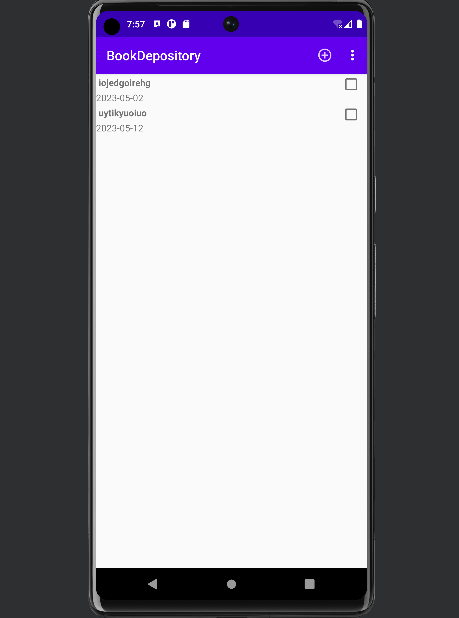
fragment\_book\_list.xml

<?xml version="1.0" encoding="utf-8"?>  
<androidx.recyclerview.widget.RecyclerView android:id="@+id/book\_recycler\_view"  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 xmlns:tools="http://schemas.android.com/tools">  
</androidx.recyclerview.widget.RecyclerView>

list\_item\_book.xml

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
  
 xmlns:tools="http://schemas.android.com/tools">  
 <CheckBox  
 android:id="@+id/list\_item\_book\_readed\_check\_box"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentRight="true" android:padding="4dp"/>  
  
 <TextView  
 android:id="@+id/list\_item\_book\_title\_text\_view"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_toLeftOf= "@id/list\_item\_book\_readed\_check\_box"  
 android:textStyle="bold"  
 android:padding="4dp"  
 android:text="Заголовок книги"/>  
  
 <TextView  
 android:id="@+id/list\_item\_book\_date\_text\_view"  
 android:layout\_width="match\_parent" android:layout\_height="wrap\_content"  
 android:layout\_toLeftOf= "@id/list\_item\_book\_readed\_check\_box"  
 android:layout\_below="@id/list\_item\_book\_title\_text\_view"  
 android:text="Дата прочтения книги"  
 />  
  
  
</RelativeLayout>

Результат



ТЕМА 10. СОЗДАНИЕ ПРОЦЕДУР ОБРАБОТКИ ИНФОРМАЦИИ

Book.java, BookLab.java, BookListActivity остались без изменений в том числе и макеты xml.

BookActivity.java

package ru.rsue.leynert.bookdepository;

import android.content.Context;  
import android.content.Intent;  
import android.os.Bundle;  
import android.support.v4.app.Fragment;  
import android.support.v4.app.FragmentActivity;  
import android.support.v4.app.FragmentManager;  
  
import java.util.UUID;  
  
public class BookActivity extends SingleFragmentActivity{  
 private static final String *EXTRA\_BOOK\_ID* = "com.example.bookdepository.book\_id";  
 public static Intent newIntent(Context packageContext, UUID bookId) {  
 Intent intent = new Intent(packageContext, BookActivity.class); intent.putExtra(*EXTRA\_BOOK\_ID*, bookId);  
 return intent;  
  
 }  
  
 @Override  
 protected Fragment createFragment() {  
  
 UUID bookId = (UUID) getIntent()  
 .getSerializableExtra(*EXTRA\_BOOK\_ID*);  
 return BookFragment2.*newInstance*(bookId);  
  
 }  
  
}

BookFragment.java

package ru.rsue.leynert.bookdepository;

import java.util.Date;  
import java.util.UUID;  
  
import android.os.Bundle;  
import android.support.v4.app.Fragment;  
import android.text.Editable;  
import android.text.TextWatcher;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.Button;  
import android.widget.CheckBox;  
import android.widget.CompoundButton;  
import android.widget.EditText;  
  
public class BookFragment2 extends Fragment {  
 private static final String *ARG\_BOOK\_ID* = "book\_id";  
 private Book mBook;  
 private EditText mTitleField;  
 private Button mDateButton;  
 private CheckBox mReadedCheckBox;  
 public static BookFragment2 newInstance(UUID bookId) {  
 Bundle args = new Bundle();  
 args.putSerializable(*ARG\_BOOK\_ID*, bookId);  
 BookFragment2 fragment = new BookFragment2();  
 fragment.setArguments(args);  
 return fragment;  
 }  
  
 @Override  
 public void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 UUID bookId = (UUID) getArguments().getSerializable(*ARG\_BOOK\_ID*);  
 mBook = BookLab.*get*(getActivity()).getBook(bookId);  
  
 }  
 @Override  
 public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {  
 View v = inflater.inflate(R.layout.*fragment\_book*, container, false);  
 mTitleField = (EditText) v.findViewById(R.id.*book\_title*);  
 mTitleField.setText(mBook.getTitle());  
 mTitleField.addTextChangedListener(new TextWatcher() {  
 @Override  
 public void beforeTextChanged(CharSequence s, int start, int count, int after) {  
  
 }  
  
 @Override  
 public void onTextChanged(CharSequence s, int start, int before, int count) {  
 mBook.setTitle(s.toString());  
 }  
  
 @Override  
 public void afterTextChanged(Editable c) {  
  
 }  
 });  
 mDateButton = (Button) v.findViewById(R.id.*book\_date*);  
 mDateButton.setText(mBook.getDate().toString());  
 mDateButton.setEnabled(false);  
mReadedCheckBox = (CheckBox)v.findViewById(R.id.*book\_readed*);  
 mReadedCheckBox.setChecked(mBook.isReaded());  
 mReadedCheckBox.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {  
 @Override  
 public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {  
 mBook.setReaded(isChecked);  
 }  
});  
 return v;

BookListFragment.java

package ru.rsue.leynert.bookdepository;

import android.content.Intent;  
import android.os.Bundle;  
import android.support.v4.app.Fragment;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.CheckBox;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import androidx.annotation.NonNull;  
import androidx.recyclerview.widget.LinearLayoutManager;  
import androidx.recyclerview.widget.RecyclerView;  
  
import java.util.List;  
  
public class BookListFragment extends Fragment {  
 private RecyclerView mBookRecyclerView;  
 private BookAdapter mAdapter;  
 @Override  
 public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {  
 View view = inflater.inflate(R.layout.*fragment\_book\_list*, container, false);  
 mBookRecyclerView = (RecyclerView) view  
 .findViewById(R.id.*book\_recycler\_view*);  
 mBookRecyclerView.setLayoutManager(new LinearLayoutManager(getActivity()));  
 updateUI();  
 return view;  
 }  
 @Override  
 public void onResume() {  
 super.onResume();  
 updateUI();  
 }  
  
 private void updateUI() {  
 BookLab bookLab = BookLab.*get*(getActivity());  
 List<Book> books = bookLab.getBooks();  
 if (mAdapter == null) {  
 mAdapter = new BookAdapter(books);  
 mBookRecyclerView.setAdapter(mAdapter);  
 } else {  
 mAdapter.notifyDataSetChanged();  
 }  
  
 }  
  
 private class BookHolder extends RecyclerView.ViewHolder implements View.OnClickListener {  
 private TextView mTitleTextView;  
 private TextView mDateTextView;  
 private Book mBook;  
 private CheckBox mReadedCheckBox;  
  
 public void bindBook(Book book) { mBook = book;  
 mTitleTextView.setText(mBook.getTitle());  
 mDateTextView.setText(mBook.getDate().toString());  
 mReadedCheckBox.setChecked(mBook.isReaded());  
 }  
 public BookHolder(View itemView) {  
 super(itemView);  
 itemView.setOnClickListener(this);  
 mTitleTextView = (TextView)  
 itemView.findViewById(R.id.*list\_item\_book\_title\_text\_view*); mDateTextView = (TextView)  
 itemView.findViewById(R.id.*list\_item\_book\_date\_text\_view*); mReadedCheckBox = (CheckBox)  
 itemView.findViewById(R.id.*list\_item\_book\_readed\_check\_box*);  
  
 }  
 @Override  
 public void onClick(View v) {  
 Intent intent = BookActivity.*newIntent*(getActivity(), mBook.getId());  
 startActivity(intent);  
 }  
}  
  
  
  
 private class BookAdapter extends RecyclerView.Adapter<BookHolder> {  
 private List<Book> mBooks;  
 public BookAdapter(List<Book> books) {  
 mBooks = books;  
 }  
  
 @Override  
 public BookHolder onCreateViewHolder( ViewGroup parent, int viewType) {  
 LayoutInflater layoutInflater = LayoutInflater.*from*(getActivity());  
 View view = layoutInflater.inflate(R.layout.*list\_item\_book*, parent, false);  
 return new BookHolder(view);  
  
 }  
  
 @Override  
 public void onBindViewHolder(BookHolder holder, int position) {  
 Book book = mBooks.get(position);  
 holder.bindBook(book);  
 }  
  
 @Override  
 public int getItemCount() {  
  
 return mBooks.size();  
 }  
 }  
  
}

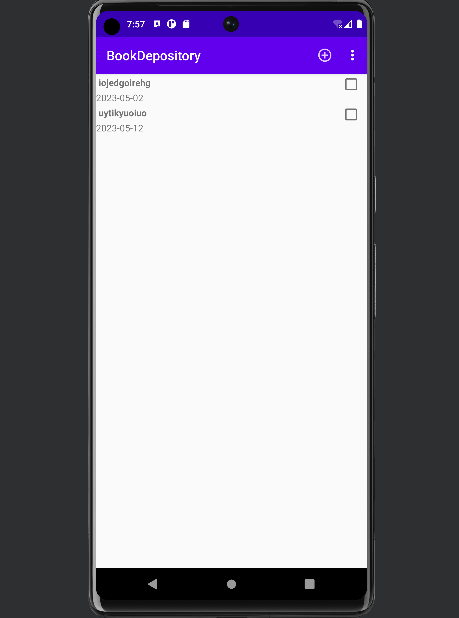
SingleFragmentActivity.java

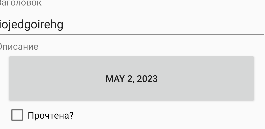
package ru.rsue.leynert.bookdepository;

import android.os.Bundle;  
import android.support.v4.app.Fragment;  
import android.support.v4.app.FragmentActivity;  
import android.support.v4.app.FragmentManager;  
  
public abstract class SingleFragmentActivity extends FragmentActivity {  
 protected abstract Fragment createFragment();  
  
 @Override  
 public void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_fragment*);  
 FragmentManager fm = getSupportFragmentManager();  
 Fragment fragment = fm.findFragmentById(R.id.*fragmentContainer*);  
 if (fragment == null) {  
 fragment = createFragment();//BookFragment2  
 fm.beginTransaction()  
 .add(R.id.*fragmentContainer*, fragment)  
 .commit();  
 }  
 }  
}

Результат

У нас есть список книг, выбираем любую из них и смотрим подробную информацию.





ТЕМА 11. ОБЕСПЕЧЕНИЕ ДОПОЛНИТЕЛЬНОЙ ФУНКЦИОНАЛЬНОЙ СИСТЕМЫ

Создал новый класс

BookPagerActivity.java

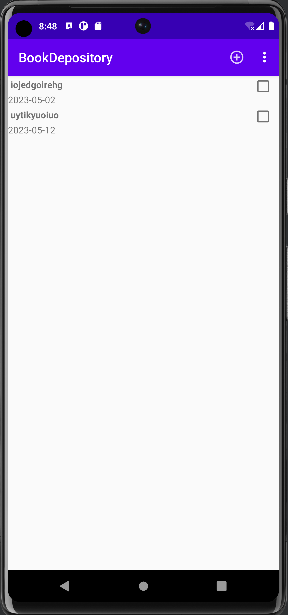
package com.example.bookdepository;  
  
import android.content.Context;  
import android.content.Intent;  
import android.os.Bundle;  
import android.support.v4.app.Fragment;  
import android.support.v4.app.FragmentActivity;  
import android.support.v4.app.FragmentManager;  
import android.support.v4.app.FragmentStatePagerAdapter;  
import android.support.v4.view.ViewPager;  
  
import java.util.List;  
import java.util.UUID;  
  
public class BookPagerActivity extends FragmentActivity {  
 private static final String *EXTRA\_BOOK\_ID* = "com.example.bookdepository.book\_id";  
 private ViewPager mViewPager;  
 private List<Book> mBooks;  
 public static Intent newIntent(Context packageContext, UUID bookId)  
 {  
 Intent intent = new Intent(packageContext, BookPagerActivity.class);  
 intent.putExtra(*EXTRA\_BOOK\_ID*, bookId); return intent;  
 }  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_book\_pager*);  
 UUID bookId = (UUID) getIntent()  
 .getSerializableExtra(*EXTRA\_BOOK\_ID*);  
 mViewPager = (ViewPager) findViewById( R.id.*activity\_book\_pager\_view\_pager*);  
 mBooks = BookLab.*get*(this).getBooks();  
 FragmentManager fragmentManager = getSupportFragmentManager();  
 mViewPager.setAdapter(new FragmentStatePagerAdapter(fragmentManager) {  
 @Override  
 public Fragment getItem(int position) {  
 Book book = mBooks.get(position);  
 return BookFragment2.*newInstance*(book.getId());  
  
 }  
  
 @Override  
 public int getCount() {  
 return mBooks.size();  
 }  
 });  
 for (int i = 0; i < mBooks.size(); i++) {  
 if (mBooks.get(i).getId().equals(bookId)) {  
 mViewPager.setCurrentItem(i);  
 break;  
 }  
 }  
  
 }

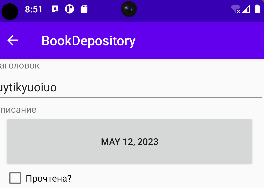
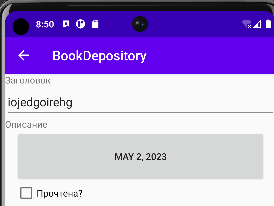
Удалил класс BookActivity.java

Добавила макет activity\_book\_pager.xml

<?xml version="1.0" encoding="utf-8"?>  
<android.support.v4.view.ViewPager xmlns:android="http://schemas.android.com/apk/res/android"  
 android:id = "@+id/activity\_book\_pager\_view\_pager"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
</android.support.v4.view.ViewPager>

Результат стал список книг, по которым мы выбираем какую-либо книгу и можем перелистывать с одной книги, с помощью продвижение экрана влево или вправо.





ТЕМА 12. INTENTS

<string name="book\_readed\_label">Прочтена?</string>  
<string name="date\_picker\_title">Дата начала прочтения:</string>

DatePickerFragment.java

package ru.rsue.leynert.bookdepository;  
  
import android.app.Activity;  
import android.app.AlertDialog;  
import android.app.Dialog;  
import android.content.DialogInterface;  
import android.content.Intent;  
import android.os.Bundle;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.widget.DatePicker;  
  
import androidx.fragment.app.DialogFragment;  
  
import java.util.Calendar;  
import java.util.Date;  
import java.util.GregorianCalendar;  
  
  
  
public class DatePickerFragment extends DialogFragment {  
  
 private static final String *ARG\_DATE* = "date";  
 private DatePicker mDatePicker;  
 public static final String *EXTRA\_DATE* =  
 "ru.rsue.bugakovnikita.bookdepository.date";  
 public static DatePickerFragment newInstance(Date date) {  
 Bundle args = new Bundle();  
 args.putSerializable(*ARG\_DATE*, date);  
  
 DatePickerFragment fragment = new DatePickerFragment();  
 fragment.setArguments(args);  
 return fragment;  
 }  
  
  
 @Override  
 public Dialog onCreateDialog(Bundle savedInstanceState) {  
 Date date = (Date) getArguments().getSerializable(*ARG\_DATE*);  
 Calendar calendar = Calendar.*getInstance*();  
 calendar.setTime(date);  
 View v = LayoutInflater.*from*(getActivity())  
 .inflate(R.layout.*dialog\_date*, null);  
 int year = calendar.get(Calendar.*YEAR*);  
 int month = calendar.get(Calendar.*MONTH*);  
 int day = calendar.get(Calendar.*DAY\_OF\_MONTH*);  
 mDatePicker = (DatePicker)  
 v.findViewById(R.id.*dialog\_date\_date\_picker*);  
 mDatePicker.init(year, month, day, null);  
 return new AlertDialog.Builder(getActivity())  
 .setView(v)  
 .setTitle(R.string.*date\_picker\_title*)  
 .setPositiveButton(android.R.string.*ok*,new DialogInterface.OnClickListener() {  
 @Override  
 public void onClick(DialogInterface dialog, int which) {  
 int year = mDatePicker.getYear();  
 int month = mDatePicker.getMonth();  
 int day = mDatePicker.getDayOfMonth();  
 Date date = new GregorianCalendar(year, month, day).  
 getTime();  
 sendResult(Activity.*RESULT\_OK*, date);  
 }  
 } ).create();  
  
  
 }  
 private void sendResult(int resultCode, Date date) {  
 if (getTargetFragment() == null) {  
 return;  
 }  
 Intent intent = new Intent();  
 intent.putExtra(*EXTRA\_DATE*, date);  
 getTargetFragment()  
 .onActivityResult(getTargetRequestCode(), resultCode,  
 intent);  
 }  
  
}

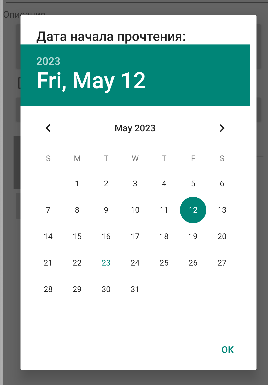
BookFragment.java

private static final String *DIALOG\_DATE* = "DialogDate";

mDateButton.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 FragmentManager manager = getFragmentManager();  
 DatePickerFragment dialog = DatePickerFragment  
 .*newInstance*(mBook.getDate());  
 dialog.setTargetFragment(BookFragment.this,  
 *REQUEST\_DATE*);  
 dialog.show(manager, *DIALOG\_DATE*);  
 }  
});

Dialog\_date.xml

<?xml version="1.0" encoding="utf-8"?>  
<android.widget.DatePicker xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:id="@+id/dialog\_date\_date\_picker">  
</android.widget.DatePicker>



ТЕМА 13. ТЕСТИРОВАНИЕ ПРИЛОЖЕНИЯ

AndroidManifest.xml

<application  
 android:allowBackup="true"  
 android:dataExtractionRules="@xml/data\_extraction\_rules"  
 android:fullBackupContent="@xml/backup\_rules"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:supportsRtl="true"  
 android:theme="@style/Theme.BookDepository"

SingleFragmentActivity.java



BookPagerActivity.java



Strings.xml

<string name="date\_picker\_title">Дата начала прочтения:</string>  
<string name="new\_book">Новая книга</string>  
<string name="show\_subtitle">Показать подзаголовок</string>  
<string name="hide\_subtitle">Скрыть подзаголовок</string>  
<string name="subtitle\_format">%1$s books</string>



Menu/fragment\_book\_list.xml

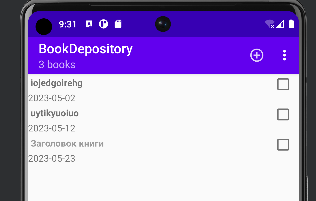
<?xml version="1.0" encoding="utf-8"?>  
<menu xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto">  
 <item  
 android:id="@+id/menu\_item\_new\_book"  
 android:icon="@drawable/ic\_menu\_add"  
 android:title="@string/new\_book"  
 app:showAsAction="ifRoom|withText"/>  
 <item  
 android:id="@+id/menu\_item\_show\_subtitle"  
 android:title="@string/show\_subtitle"  
 app:showAsAction="ifRoom"/>  
</menu>

BookListFragment.java

@Override  
public void onCreateOptionsMenu(Menu menu, MenuInflater inflater) {  
 super.onCreateOptionsMenu(menu, inflater);  
 inflater.inflate(R.menu.*fragment\_book\_list*, menu);  
  
 MenuItem subtitleItem = menu.findItem(R.id.*menu\_item\_show\_subtitle*);  
 if (mSubtitleVisible) {  
 subtitleItem.setTitle(R.string.*hide\_subtitle*);  
 } else {  
 subtitleItem.setTitle(R.string.*show\_subtitle*);  
 }  
}

BookListFragment.java

@Override  
public void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setHasOptionsMenu(true);  
}



BookListFragment.java

private void updateSubtitle() {  
 BookLab bookLab = BookLab.*get*(getActivity());  
 int bookCount = bookLab.getBooks().size();  
 String subtitle = getString(R.string.*subtitle\_format*, bookCount);  
 if (!mSubtitleVisible) {  
 subtitle = null;  
 }  
 AppCompatActivity activity = (AppCompatActivity) getActivity();  
 activity.getSupportActionBar().setSubtitle(subtitle);  
}

@Override  
public boolean onOptionsItemSelected(MenuItem item) {  
 switch (item.getItemId()) {  
 case R.id.*menu\_item\_new\_book*:  
 Book book = new Book(UUID.*randomUUID*());  
 BookLab.*get*(getActivity()).addBook(book);  
 Intent intent = BookPagerActivity  
 .*newIntent*(getActivity(), book.getId());  
 startActivity(intent);  
 case R.id.*menu\_item\_show\_subtitle*:  
 mSubtitleVisible = !mSubtitleVisible;  
 getActivity().invalidateOptionsMenu();  
 updateSubtitle();  
 return true;  
 default:  
 return super.onOptionsItemSelected(item);  
 }  
}

MenuItem subtitleItem = menu.findItem(R.id.*menu\_item\_show\_subtitle*);  
 if (mSubtitleVisible) {  
 subtitleItem.setTitle(R.string.*hide\_subtitle*);  
 } else {  
 subtitleItem.setTitle(R.string.*show\_subtitle*);  
 }  
}

@Override  
public void onSaveInstanceState(Bundle outState) {  
 super.onSaveInstanceState(outState);  
 outState.putBoolean(*SAVED\_SUBTITLE\_VISIBLE*, mSubtitleVisible);  
}



ТЕМА 14. ОТЛАДКА ПРИЛОЖЕНИЯ

BookBaseWrapper.java

package ru.rsue.leynert.bookdepository.database;  
  
import android.content.Context;  
import android.database.sqlite.SQLiteDatabase;  
import android.database.sqlite.SQLiteOpenHelper;  
  
  
public class BookBaseHelper extends SQLiteOpenHelper {  
  
 private static final int *VERSION* = 1;  
 private static final String *DATABASE\_NAME* = "bookBase.db";  
  
 public BookBaseHelper(Context context) {  
 super(context, *DATABASE\_NAME*, null, *VERSION*);  
 }  
  
 @Override  
 public void onCreate(SQLiteDatabase db) {  
 db.execSQL("create table " + BookDbSchema.BookTable.*NAME* + "(" +  
 " \_id integer primary key autoincrement, " +  
 BookDbSchema.BookTable.Cols.*UUID* + ", " +  
 BookDbSchema.BookTable.Cols.*TITLE* + ", " +  
 BookDbSchema.BookTable.Cols.*DATE* + ", " +  
 BookDbSchema.BookTable.Cols.*READED* + ")"  
 );  
 }  
 @Override  
 public void onUpgrade (SQLiteDatabase db,int i, int i1){  
  
 }  
  
}

BookCursorWrapper.java

package ru.rsue.leynert.bookdepository.database;  
  
import android.database.Cursor;  
import android.database.CursorWrapper;  
  
import java.sql.Date;  
import java.util.UUID;  
  
import ru.rsue.leynert.bookdepository.Book;  
  
  
public class BookCursorWrapper extends CursorWrapper {  
 public BookCursorWrapper(Cursor cursor) {  
 super(cursor);  
 }  
 public Book getBook() {  
 String uuidString =  
 getString(getColumnIndex(BookDbSchema.BookTable.Cols.*UUID*));  
 String title = getString(getColumnIndex(BookDbSchema.BookTable.Cols.*TITLE*));  
 long date = getLong(getColumnIndex(BookDbSchema.BookTable.Cols.*DATE*));  
 int isReaded = getInt(getColumnIndex(BookDbSchema.BookTable.Cols.*READED*));  
 Book book = new Book(UUID.*fromString*(uuidString));  
 book.setTitle(title);  
 book.setDate(new Date(date));  
 book.setReaded(isReaded != 0);  
 return book;  
 }  
}

BookDbSchema.java

package ru.rsue.leynert.bookdepository.database;  
  
import android.database.Cursor;  
import android.database.CursorWrapper;  
  
import java.sql.Date;  
import java.util.UUID;  
  
import ru.rsue.leynert.bookdepository.Book;  
  
  
public class BookCursorWrapper extends CursorWrapper {  
 public BookCursorWrapper(Cursor cursor) {  
 super(cursor);  
 }  
 public Book getBook() {  
 String uuidString =  
 getString(getColumnIndex(BookDbSchema.BookTable.Cols.*UUID*));  
 String title = getString(getColumnIndex(BookDbSchema.BookTable.Cols.*TITLE*));  
 long date = getLong(getColumnIndex(BookDbSchema.BookTable.Cols.*DATE*));  
 int isReaded = getInt(getColumnIndex(BookDbSchema.BookTable.Cols.*READED*));  
 Book book = new Book(UUID.*fromString*(uuidString));  
 book.setTitle(title);  
 book.setDate(new Date(date));  
 book.setReaded(isReaded != 0);  
 return book;  
 }  
}

ТЕМА 15. ЗАЩИТА ПРИЛОЖЕНИЯ.ОЦЕНКА КАЧЕСТВА И НАДЕЖНОСТИ ПРИЛОЖЕНИЯ.

Strings.xml

<string name="book\_report\_text">Отправить отчет о прочтении</string>  
<string name="book\_report">%1$s Книга была запланирована к прочтению %2$s и %3$s</string>  
<string name="book\_report\_readed">Книга прочитана</string>  
<string name="book\_report\_unreaded">Книга не прочитана</string>  
<string name="book\_report\_subject">BookDepository отчет о прочтении</string>  
<string name="send\_report">Отправить отчет о прочтении через</string>

Fragment\_book.xml

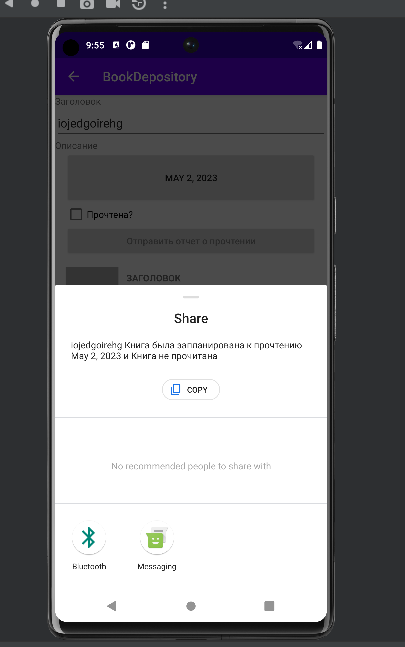
<Button  
 android:id="@+id/book\_report"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="16dp"  
 android:layout\_marginRight="16dp"  
 android:hint="@string/book\_report\_text" />

BookFragment.java

private Button mReportButton;

mReportButton = (Button)v.findViewById(R.id.*book\_report*);  
mReportButton.setOnClickListener(new View.OnClickListener() {  
 public void onClick(View v) {  
 Intent i = new Intent(Intent.*ACTION\_SEND*);  
 i.setType("text/plain");  
 i.putExtra(Intent.*EXTRA\_TEXT*, getBookReport());  
 i.putExtra(Intent.*EXTRA\_SUBJECT*,  
 getString(R.string.*book\_report\_subject*));  
 i = Intent.*createChooser*(i, getString(R.string.*send\_report*));  
 startActivity(i);  
 }  
});

private String getBookReport() {  
 String readedString = null;  
 if (mBook.isReaded()){  
 readedString = getString(R.string.*book\_report\_readed*);  
 }else {  
 readedString = getString(R.string.*book\_report\_unreaded*);  
 }  
 String dateFormat = "EEE, MMM dd";  
 String dateString = DateFormat  
 .*getDateInstance*(DateFormat.*MEDIUM*).format(mBook.getDate());  
 String report = getString(R.string.*book\_report*,  
 mBook.getTitle(), dateString, readedString);  
 return report;  
}



ТЕМА 16. СОПРОЖДЕНИЕ ПРОГРАММ

View\_camera\_and\_title.xml

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="16dp"  
 android:layout\_marginRight="16dp"  
 android:layout\_marginTop="16dp">  
 <LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 android:layout\_marginRight="4dp">  
 <ImageView  
 android:id="@+id/book\_photo"  
 android:layout\_width="80dp"  
 android:layout\_height="80dp"  
 android:scaleType="centerInside"  
 android:background="@android:color/darker\_gray"  
 android:cropToPadding="true"/>  
 <ImageButton  
 android:id="@+id/book\_camera"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:src="@android:drawable/ic\_menu\_camera"/>  
 </LinearLayout>  
 <LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 android:layout\_weight="1">  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="@string/book\_title\_label"  
 style="?android:listSeparatorTextViewStyle"/>  
 <EditText  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/book\_title"  
 android:layout\_marginRight="16dp"  
 android:hint="@string/book\_title\_hint"/>  
</LinearLayout>  
</LinearLayout>

Fragment\_book.xml

<include  
 layout="@layout/view\_camera\_and\_title"/>

Book.java

public String getPhotoFilename() {  
 return "IMG\_" + getId().toString() + ".jpg";  
}

BookLab.java

public File getPhotoFile(Book book) {  
 File externalFilesDir = mContext  
 .getExternalFilesDir(Environment.*DIRECTORY\_PICTURES*);  
 if (externalFilesDir == null) {  
 return null;  
 }  
 return new File(externalFilesDir, book.getPhotoFilename());  
}

BookFragment.java

private File mPhotoFile;

private ImageButton mPhotoButton;  
private ImageView mPhotoView;  
  
private static final int *REQUEST\_DATE* = 0;  
private static final int *REQUEST\_PHOTO* = 1;

@Override  
public void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 UUID bookId = (UUID) getArguments().getSerializable(*ARG\_BOOK\_ID*);  
 mBook = BookLab.*get*(getActivity()).getBook(bookId);  
 mPhotoFile = BookLab.*get*(getActivity()).getPhotoFile(mBook);  
}

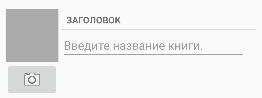
mPhotoButton = (ImageButton) v.findViewById(R.id.*book\_camera*);  
 final Intent captureImage =  
 new Intent(MediaStore.*ACTION\_IMAGE\_CAPTURE*);  
 PackageManager packageManager = getActivity().getPackageManager();  
 boolean canTakePhoto = mPhotoFile != null ||  
 captureImage.resolveActivity(packageManager) != null;  
 if (canTakePhoto) {  
 Uri uri;  
 if (Build.VERSION.*SDK\_INT*<24)  
 uri = Uri.*fromFile*(mPhotoFile);  
 else  
 uri = FileProvider.*getUriForFile*(getActivity(),  
 BuildConfig.*APPLICATION\_ID* + ".provider", mPhotoFile);  
 captureImage.putExtra(MediaStore.*EXTRA\_OUTPUT*, uri);  
 mPhotoButton.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 startActivityForResult(captureImage, *REQUEST\_PHOTO*);  
 }  
 });  
 mPhotoView = (ImageView) v.findViewById(R.id.*book\_photo*);  
 updatePhotoView();  
 }

AndroidManifest.java

<provider  
 android:name="androidx.core.content.FileProvider"  
 android:authorities="${applicationId}.provider"  
 android:exported="false"  
 android:grantUriPermissions="true">  
 <meta-data  
 android:name="android.support.FILE\_PROVIDER\_PATHS"  
 android:resource="@xml/provider\_paths"/>  
</provider>

Provider\_paths.xml

<?xml version="1.0" encoding="utf-8"?>  
<paths xmlns:android="http://schemas.android.com/apk/res/android">  
 <external-path name="external\_files" path="."/>  
</paths>





ТЕМА 17. СОПРОВОЖДЕНИЕ ПРОГРАММ

Drawable/sun.xml

<?xml version="1.0" encoding="utf-8"?>  
<shape  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 android:dither="true"  
 android:shape="oval" >  
  
 <gradient  
 android:endColor="#ffff6600"  
 android:gradientRadius="150"  
 android:startColor="#ffffcc00"  
 android:type="radial"  
 android:useLevel="false" />  
  
 <size  
 android:height="150dp"  
 android:width="150dp" />  
  
</shape>

Drawable/sky.xml

<?xml version="1.0" encoding="utf-8"?>  
<shape  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 android:dither="true"  
 android:shape="rectangle" >  
  
 <gradient  
 android:angle="90"  
 android:endColor="#ff000033"  
 android:startColor="#ff0000ff" />  
  
</shape>

Drawable/grass.xml

<?xml version="1.0" encoding="utf-8"?>  
<shape  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 android:dither="true"  
 android:shape="rectangle" >  
  
 <gradient  
 android:angle="90"  
 android:endColor="#ff003300"  
 android:startColor="#ff009900" />  
  
</shape>

Layout/activity\_main.xml

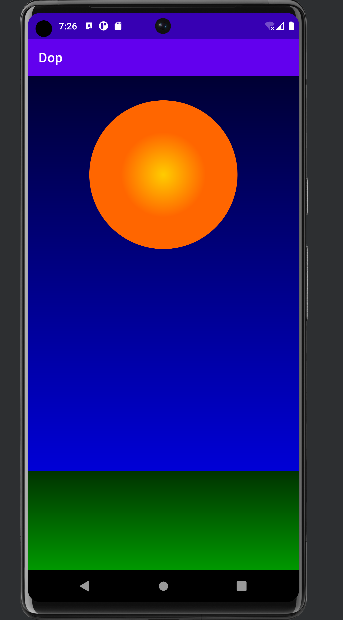
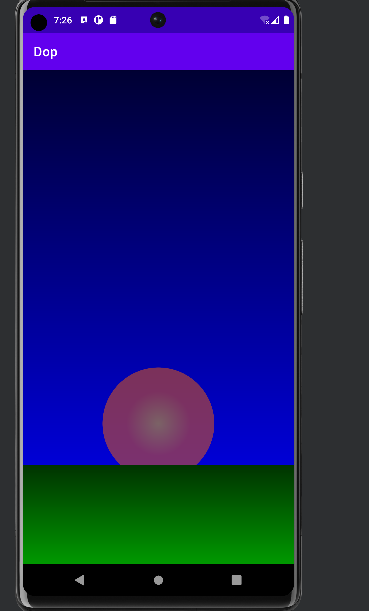
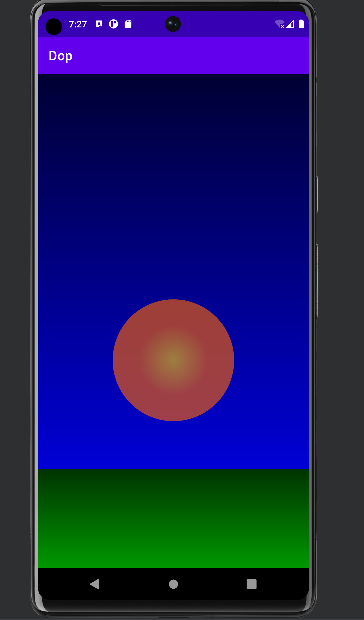
<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity" >  
  
 <ImageView  
 android:id="@+id/sky"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="fill\_parent"  
 android:contentDescription="@string/sky"  
 android:src="@drawable/sky" />  
  
 <ImageView  
 android:id="@+id/sun"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_centerHorizontal="true"  
 android:contentDescription="@string/sun"  
 android:scaleType="fitCenter"  
 android:src="@drawable/sun" />  
  
 <ImageView  
 android:id="@+id/grass"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="150dp"  
 android:layout\_alignParentBottom="true"  
 android:contentDescription="@string/grass"  
 android:src="@drawable/grass" />  
  
</RelativeLayout>

Anim/sun\_rise.xml

<?xml version="1.0" encoding="utf-8"?>  
<set  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 android:duration="5000"  
 android:fillAfter="true"  
 android:interpolator="@android:anim/accelerate\_decelerate\_interpolator"  
 android:shareInterpolator="false" >  
 <scale  
 android:fromXScale="1.0"  
 android:fromYScale="1.0"  
 android:pivotX="50%"  
 android:pivotY="50%"  
 android:toXScale="1.5"  
 android:toYScale="1.5" />  
 <translate android:fromYDelta="80%p"  
 android:toYDelta="10%p" />  
 <alpha  
 android:fromAlpha="0.3"  
 android:toAlpha=" 1.0" />  
</set>

MainActivity.java

package ru.rsue.dop;  
  
import android.os.Bundle;  
import android.view.animation.Animation;  
import android.view.animation.AnimationUtils;  
import android.widget.ImageView;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
public class MainActivity extends AppCompatActivity {  
 @Override  
 protected void onCreate(Bundle savedlnstanceState) {  
 super.onCreate(savedlnstanceState);  
 setContentView(R.layout.*activity\_main*);  
// Получаем ссылку на изображение солнца  
 ImageView sunImageView = (ImageView)  
 findViewById(R.id.*sun*);  
// Анимация для восхода солнца Animation  
 Animation sunRiseAnimation = AnimationUtils.*loadAnimation*(this, R.anim.*sun\_rise*);  
 // Подключаем анимацию к нужному View  
 sunImageView.startAnimation(sunRiseAnimation);  
 }  
}



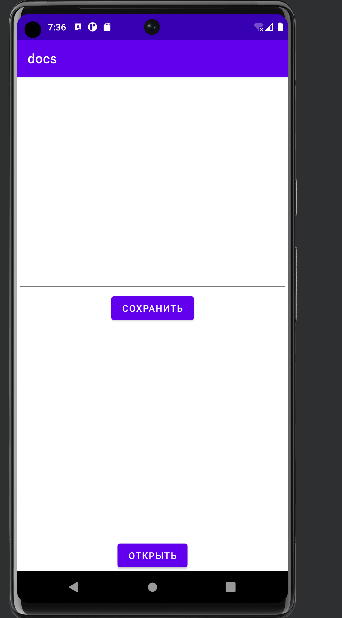
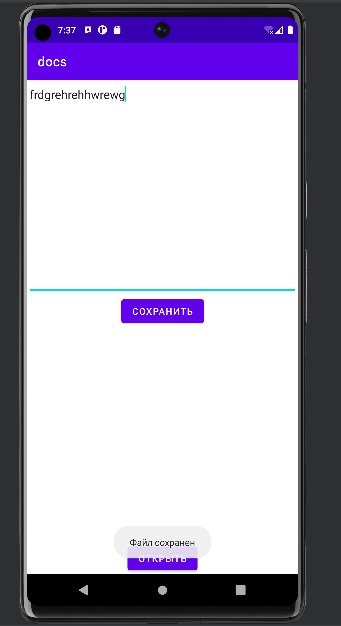
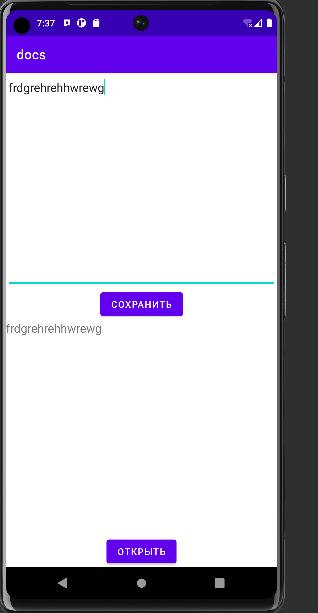
ТЕМА 18. ИТОГОВОЕ ЗАНЯТИЕ

Layout/activity\_main.xml

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
 <EditText  
 android:id="@+id/editor"  
 android:layout\_width="0dp"  
 android:layout\_height="0dp"  
 android:textSize="18sp"  
 android:gravity="start"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintBottom\_toTopOf="@id/save\_text"  
 app:layout\_constraintTop\_toTopOf="parent" />  
 <Button  
 android:id="@+id/save\_text"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:onClick="saveText"  
 android:text="Сохранить"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintBottom\_toTopOf="@id/text"  
 app:layout\_constraintTop\_toBottomOf="@id/editor" />  
 <TextView  
 android:id="@+id/text"  
 android:layout\_width="0dp"  
 android:layout\_height="0dp"  
 android:gravity="start"  
 android:textSize="18sp"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintBottom\_toTopOf="@+id/open\_text"  
 app:layout\_constraintTop\_toBottomOf="@+id/save\_text"  
 />  
 <Button  
 android:id="@+id/open\_text"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:onClick="openText"  
 android:text="Открыть"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/text" />  
</androidx.constraintlayout.widget.ConstraintLayout>

MainActivity.java

package ru.rsue.docs;  
  
import androidx.appcompat.app.AppCompatActivity;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.EditText;  
import android.widget.TextView;  
import android.widget.Toast;  
import java.io.FileInputStream;  
import java.io.FileOutputStream;  
import java.io.IOException;  
public class MainActivity extends AppCompatActivity {  
 private final static String *FILE\_NAME* = "content.txt";  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 }  
 // сохранение файла  
 public void saveText(View view){  
 FileOutputStream fos = null;  
 try {  
 EditText textBox = (EditText)  
 findViewById(R.id.*editor*);  
 String text = textBox.getText().toString();  
 fos = openFileOutput(*FILE\_NAME*, *MODE\_PRIVATE*);  
 fos.write(text.getBytes());  
 Toast.*makeText*(this, "Файл сохранен",  
 Toast.*LENGTH\_SHORT*).show();  
 }  
 catch(IOException ex) {  
 Toast.*makeText*(this, ex.getMessage(),  
 Toast.*LENGTH\_SHORT*).show();  
 }  
 finally{  
 try{  
 if(fos!=null)  
 fos.close();  
 }  
 catch(IOException ex){  
  
 Toast.*makeText*(this, ex.getMessage(),  
 Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
 }  
 // открытие файла  
 public void openText(View view){  
 FileInputStream fin = null;  
 TextView textView = (TextView)  
 findViewById(R.id.*text*);  
 try {  
 fin = openFileInput(*FILE\_NAME*);  
 byte[] bytes = new byte[fin.available()];  
 fin.read(bytes);  
 String text = new String (bytes);  
 textView.setText(text);  
 }  
 catch(IOException ex) {  
 Toast.*makeText*(this, ex.getMessage(),  
 Toast.*LENGTH\_SHORT*).show();  
 }  
 finally{  
 try{  
 if(fin!=null)  
 fin.close();  
 }  
 catch(IOException ex){  
 Toast.*makeText*(this, ex.getMessage(),  
 Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
 }  
}



DroidQuest

<https://github.com/rixie13/DroidQuest.git>

BookDepository

<https://github.com/rixie13/BookDepository.git>

Sunrise

<https://github.com/rixie13/sunrise.git>

Docs

<https://github.com/rixie13/docs.git>