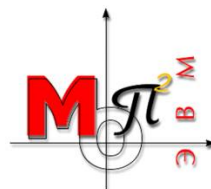


МИНОБРНАУКИ РОССИИ

Федеральное государственное автономное образовательное
учреждение высшего образования
«ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт компьютерных технологий и информационной безопасности
Кафедра Математического обеспечения и применения ЭВМ



ОТЧЁТ

по лабораторной работе №2
по курсу «ПОМП и СЦОС»

Выполнила:

студентка группы КТмо1-3
Куприянова А.А.

Проверила:

доцент кафедры МОП ЭВМ
Пирская Л.В.

Оценка

« ____ » _____ 2017 г.

Таганрог 2017

Оглавление

Постановка задачи.	2
Разработка приложения.	2
Описание интерфейса приложения.	5
Заключение.	1
Приложение. Листинг разработанного приложения.	1
activity_parser.xml.....	1
AvtivityParser.java.....	3
activity_service.xml	5
ActivityService.java	6

Постановка задачи.

В ходе выполнения лабораторной работы продолжается разработка приложения, позволяющего пользователю записывать собственную цитату дня, назначать цитатой дня полученную цитату с сайта citaty.info и сохранять лучшую цитату в файл.

В ходе выполнения лабораторной работы было разработано приложение с графическим интерфейсом, позволяющее пользователю записывать собственную цитату дня. Приложение содержит несколько activity, взаимодействующих с главной activity посредством передачи данных в главную activity. Также реализован фоновый сервис воспроизведения музыкальной композиции, позволяющий останавливать и запускать воспроизведение музыки.

Для реализации недостающего функционала приложения необходимо следующее:

1. Activity случайной цитаты. Приложение в фоновом режиме должно получать данные сайта citaty.info, пользователь будет иметь возможность перейти на сайт, установить полученную цитату в качестве цитаты дня, которая будет отображаться в главной activity, сохранить полученную цитату как лучшую в файл, или запросить новую случайную цитату.
2. Activity лучшей цитаты. На экран пользователя выведется цитата, сохранённая в файле как лучшая. Пользователь будет иметь возможность установить её в качестве цитаты дня.

Разработка приложения.

Получение данных с сайта выполняется с помощью экземпляра класса AsyncTask:

```
public class CiteThread extends AsyncTask<String, Void, String> {
    @Override
    protected String doInBackground(String... arg) {
        citeOfTheDay = "";
        authorOfTheDay = "";
        Document doc;
        try {
            doc = Jsoup.connect("https://citaty.info/random").get();
            cite = doc.select("#random-quote
.node__content>.field.field-name-body.field-type-text-with-summary");
            for (Element e : cite) {
                citeOfTheDay = e.text();
                if (citeOfTheDay.equals("")) {
                    break;
                }
            }
            author = doc.select("#random-quote .node__content .field-
type-taxonomy-term-reference a[title]");
            for (Element e : author) {
                String title = "";
                title = e.attr("title");
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

```

        if (title.equals("от имени автора")) {
            continue;
        }
        if ( (!title.equals("Автор
цитаты")) && (!title.equals("Цитата из книги")) && (!title.equals("")) )
            authorOfTheDay = authorOfTheDay+title+":\n";
            authorOfTheDay = authorOfTheDay+e.text()+"\n";
        }
    } catch (IOException e) {
        e.printStackTrace();
    }
    return null;
}
@Override
protected void onPostExecute(String result) {
    textCite.setText(citeOfTheDay);
    textAuthor.setText(authorOfTheDay);
}
}

```

Для извлечения необходимых данных из документа doc применяется библиотека jsoup. При помощи метода select() этой библиотеки по селекторам можно получить элементы DOM-дерева документа, притом при помощи этого метода мы получим сразу все элементы, соответствующие указанному селектору.

Из элементов, содержащих текст цитаты и данные о её источнике, при помощи метода text() получаем данные строкового типа, которые заносятся в соответствующие переменные класса при помощи метода setText.

Стоит отметить, что в случае получения текста цитаты только один элемент на странице соответствует указанному селектору. Источник цитаты в некоторых случаях представлен несколькими элементами, например: исполнитель и песня, телешоу и его участник, сериал и его персонаж и т.п. В каждом полученном методом select() элементе в атрибуте title указан тип источника цитаты, например: «Цитата из книги», «Цитата из телешоу», «Автор цитаты», «Исполнитель», «Песня», «Персонаж» и т.п.). Тип элементов получаем при помощи метода attr полученного элемента. В случаях если этот тип источника не «Цитата из книги» и «Автор цитаты», данные о типе источника добавляются к переменной источника цитаты (authorOfTheDay) вместе с указанием типа источника, в противном случае добавляется только текст информации об источнике. В результате переменная authorOfTheDay содержит наиболее полную информацию об источнике цитаты без излишней информации о типе источника.

Всё это выполняется вызове метода execute() потока, который вызывается вначале при переходе на activity, затем при нажатии на соответствующую кнопку для замены цитаты на новую случайную.

Также в этой activity предусмотрена возможность по нажатию на соответствующие кнопки установить последнюю полученную цитату в качестве цитаты дня, передав её в главную activity, аналогично тому, как это было сделано с цитатой пользователя.

Переход на сайт citaty.info также осуществляется по нажатию кнопки:

```
buttonSite.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Uri link = Uri.parse("https://citaty.info/random");
        Intent intent = new Intent(Intent.ACTION_VIEW, link);
        startActivity(intent);
    }
});
```

В случае, если пользователь решит сохранить полученную цитату как лучшую цитату в файл, обработка и запись данных осуществляется следующим образом:

```
buttonSaveTheBest.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String jstr = "";
        JSONObject jsonObj = new JSONObject();
        try {
            jsonObj.put("cite", citeOfTheDay);
            jsonObj.put("author", authorOfTheDay);
            jstr = jsonObj.toString();
        } catch (JSONException e) {
            e.printStackTrace();
        }
        try {
            OutputStreamWriter outputStreamWriter = new
            OutputStreamWriter(openFileOutput("cites.json", Context.MODE_PRIVATE));
            outputStreamWriter.write(jstr);
            outputStreamWriter.close();
        }
        catch (IOException e) {
            Log.e("Exception", "File write failed: " + e.toString());
        }
    }
});
```

Создаётся json-объект `jsonObj`, в него заносятся данные о тексте цитаты и авторе. Этот объект преобразуется в строку `jstr` методом `toString()`. При помощи экземпляра класса `OutputStreamWriter` осуществляется запись полученной из данных объекта `jsonObj` строки `jstr` в файл `cites.json`.

В activity лучшей цитаты происходит чтение из файла `cites.json` данных о тексте цитаты и авторе и вывод этих данных на экран:

```
String ret = "";

try {
    InputStream inputStream = openFileInput(FILENAME);

    if ( inputStream != null ) {
        InputStreamReader inputStreamReader = new
        InputStreamReader(inputStream);
        BufferedReader bufferedReader = new
        BufferedReader(inputStreamReader);
```

```

String receiveString = "";
StringBuilder stringBuilder = new StringBuilder();

while ( (receiveString = bufferedReader.readLine()) != null ) {
    stringBuilder.append(receiveString);
}

inputStream.close();
ret = stringBuilder.toString();
}
}
catch (FileNotFoundException e) {
    Log.e("log1", "File not found: " + e.toString());
} catch (IOException e) {
    Log.e("log2", "Can not read file: " + e.toString());
}
}
try {
    JSONObject jobj = new JSONObject(ret);
    cite = jobj.getString("cite");
    author = jobj.getString("author");
    textCite.setText(cite);
    textAuthor.setText(author);
} catch (JSONException e) {
    e.printStackTrace();
}
}

```

Чтение происходит при помощи классов `InputStreamReader` и `BufferedReader`. При помощи метода `readLine()` класса `BufferedReader` считывается очередная строка потока `inputStreamReader`. С помощью метода `toString()` класса `BufferedReader` данные преобразуются в строку, которая при помощи конструктора класса `JSONObject` преобразуется в json-объект, из которого методом `getString()` получаются данные о тексте цитаты и её источнике. Полученные данные передаются элементам разметки и отображаются на экране.

Предусмотрена также кнопка назначения полученной цитаты цитатой дня аналогично тому, как это было сделано в `activity` цитаты пользователя.

Описание интерфейса приложения.

При нажатии на главной `activity` кнопки `BEST SITE` происходит переход на `activity` лучшей цитаты. Вид `activity` показан на рисунке 1.

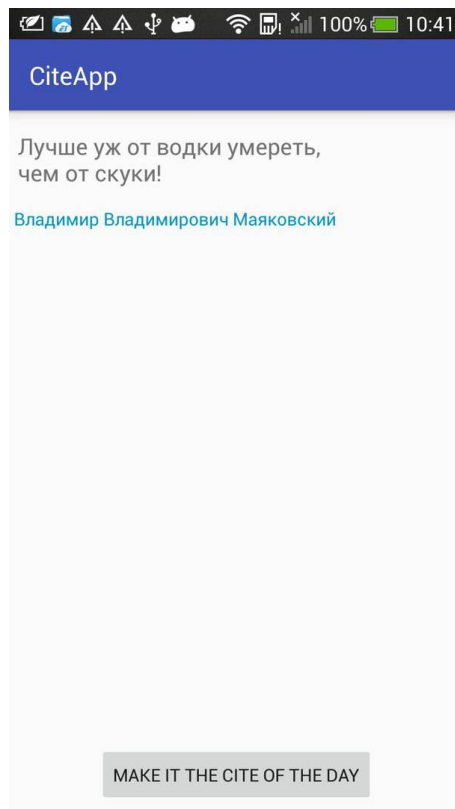


Рисунок 1 – Вид activity лучшей цитаты.

Если нажать здесь кнопку MAKE IT THE CITE OF THE DAY, произойдёт переход на главную activity, на которой будет отображаться новая установленная цитата дня (рисунок 2).



Рисунок 2. Вид главной activity после назначения лучшей цитаты цитатой дня.

Если в главной activity нажать RANDOM CITE, произойдёт переход на activity случайной цитаты, вид экрана в этом случае показан на рисунке 3.

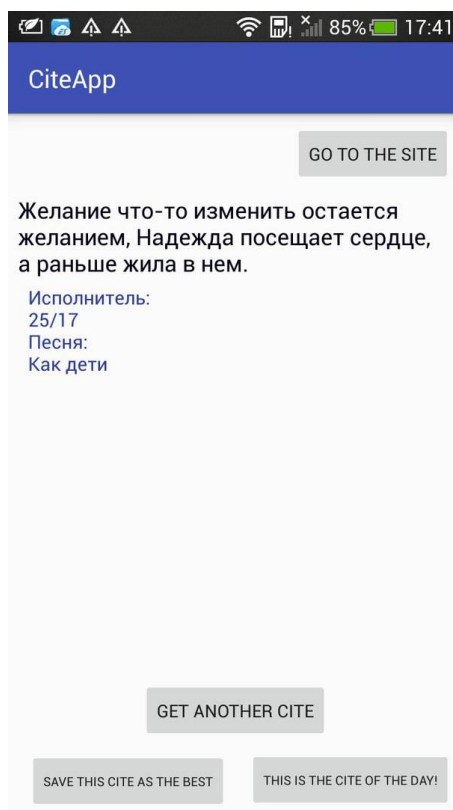


Рисунок 3 – Вид activity случайной цитаты.

Каждый раз при переходе на эту activity идёт обращение к серверу, который каждый раз выдаёт страницу с новой цитатой. Также обновление цитаты произойдёт по нажатию на кнопку GET ANOTHER CITE (рисунок 4).

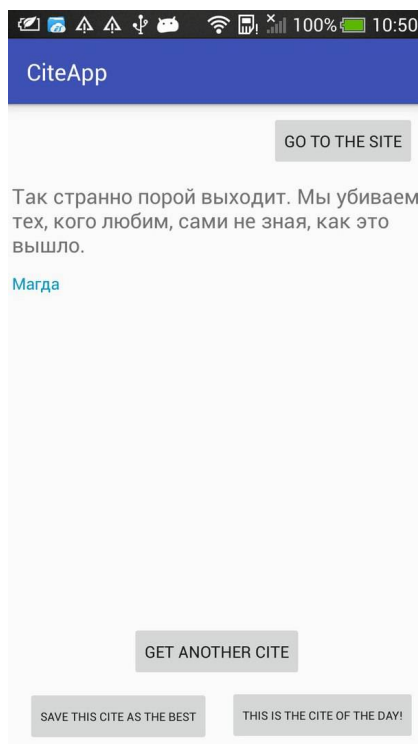


Рисунок 4 – Вид activity случайной цитаты после обновления цитаты.

Если пользователь сочтёт полученную цитату лучшей, он может сохранить её в файл, нажав на кнопку **SAVE THIS CITE AS THE BEST**. В файле всегда хранится только одна цитата, поэтому цитата, бывшая лучшей, в момент назначения новой лучшей стирается. Перейдя после этого на activity лучшей цитату сможем увидеть новую лучшую цитату на экране (рисунок 5).

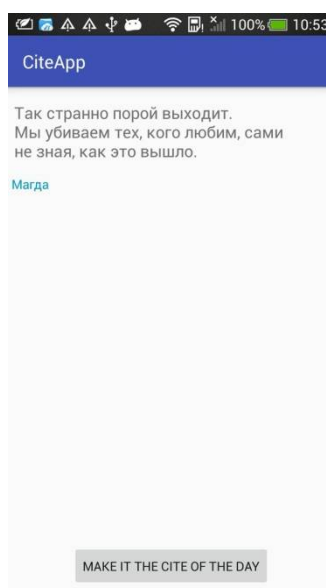


Рисунок 5 – Вид activity лучшей цитаты после её сохранения в файл в activity случайной цитаты.

В activity случайной цитаты можно не сохраняя цитату в файл назначить её цитатой дня для отображения на главной activity. Для этого необходимо снова зайти на activity случайной цитаты (рисунок 6), при этом с сайта будет получена новая цитата, и нажать кнопку THIS IS THE SITE OF THEDAY! Произойдёт переход на главную activity, где эта цитата будет отображаться как цитата дня (рисунок 7).

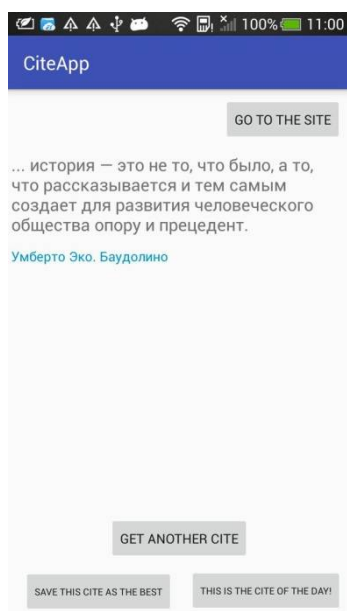


Рисунок 6. Новая цитата.

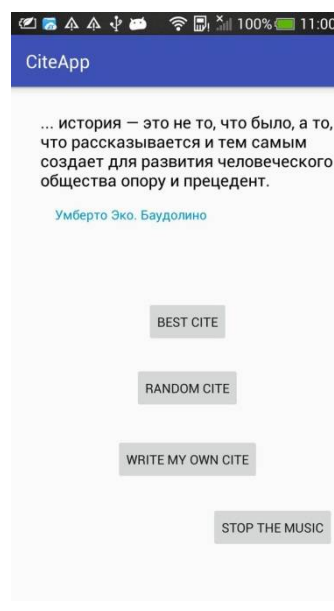


Рисунок 7. Результат назначения новой цитаты цитатой дня

Если нажать на кнопку GO TO THE SITE, необходимый сайт откроется в браузере телефона. Выглядеть он будет так, как показано на рисунке 8.

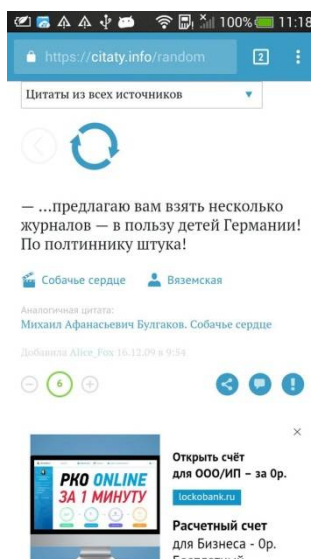


Рисунок 8 – Используемый в приложении сайт.

Заключение.

В процессе выполнения лабораторной работы разработанное ранее Android-приложение было дополнено функционалом получения данных сайта с целью извлечения оттуда цитаты и возможностью сохранения полученных данных в файл с последующим извлечением их оттуда.

Получение данных с сайта выполняется с помощью экземпляра класса AsyncTask. Для извлечения необходимых данных из документа doc применяется библиотека jsoup и её метод select() позволяющий получить элемент DOM-дерева документа по селектору. Полученные из этих элементов текстовые данные отображаются на экране в виджетах типа TextView.

Для перехода на сайт используется намерение:

```
Intent intent = new Intent(Intent.ACTION_VIEW, link),
```

где Uri link = Uri.parse("https://citaty.info/random").

Для записи данных в файл формата json использовался класс JSONObject с целью преобразования полученных строковых данных в объект. Этот объект преобразовывается в строку. Для записи данных в файл cites.json используется класс OutputStreamWriter.

Чтение из файла cites.json осуществляется с использованием классов InputStreamReader и BufferedReader (метод readLine() класса BufferedReader).

В процессе выполнения лабораторной работы я научилась получать данные с сайта в виде документа и извлекать из него необходимые данные, а также записывать в файл и считывать из него данные в формате json.

Расширить функционал приложения можно применением базы данных для хранения всех понравившихся цитат.

Приложение. Листинг разработанного приложения.

activity_parser.xml

Activity случайной цитаты. В ней поток получает данные с сайта, в ней отображается полученная с сайта цитата. Здесь располагаются кнопки перехода на сайт, получения новой случайной цитаты с сайта, назначение полученной цитаты цитатой дня с передачей данных в главную activity и сохранения полученной цитаты в файл cites.json.

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
```

```
tools:context="com.example.al.appwithserviceandparser.ActivityParser">
```

```
<TextView
    android:id="@+id/textViewCiteMain"
    android:layout_width="354dp"
    android:layout_height="wrap_content"
    android:layout_marginLeft="8dp"
    android:layout_marginRight="8dp"
    android:layout_marginTop="11dp"
    android:textSize="18sp"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/buttonSite" />

<TextView
    android:id="@+id/textViewAuthor"
    android:layout_width="353dp"
    android:layout_height="wrap_content"
    android:textAlignment="viewStart"
    android:textColor="@android:color/holo_blue_dark"
    android:textColorLink="@color/colorAccent"
    app:layout_constraintTop_toBottomOf="@+id/textViewCiteMain"
    android:layout_marginTop="12dp"
    android:layout_marginRight="8dp"
    app:layout_constraintRight_toRightOf="parent"
    android:layout_marginLeft="8dp"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintHorizontal_bias="0.533" />

<Button
    android:id="@+id/buttonGetCite"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Get another cite"
    app:layout_constraintBottom_toBottomOf="parent"
    android:layout_marginBottom="62dp"
    android:layout_marginTop="8dp"
    app:layout_constraintTop_toBottomOf="@+id/textViewAuthor"
    app:layout_constraintVertical_bias="1.0"
    android:layout_marginLeft="8dp"
    app:layout_constraintLeft_toLeftOf="parent"
    android:layout_marginRight="8dp"
    app:layout_constraintRight_toRightOf="parent" />

<Button
    android:id="@+id/buttonSite"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="go to the site"
    app:layout_constraintTop_toTopOf="parent"
    android:layout_marginTop="8dp"
    android:layout_marginRight="8dp"
    app:layout_constraintRight_toRightOf="parent" />

<Button
    android:id="@+id/buttonSaveTheBest"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="save this cite as the best"
    android:textSize="10sp"
    app:layout_constraintLeft_toLeftOf="parent"
    android:layout_marginLeft="16dp"
    android:layout_marginTop="8dp"
```

```

        app:layout_constraintTop_toBottomOf="@+id/buttonGetCite"
        app:layout_constraintBottom_toBottomOf="parent"
        android:layout_marginBottom="8dp"
        app:layout_constraintVertical_bias="0.0" />

<Button
    android:id="@+id/buttonCiteOfTheDay"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="8dp"
    android:text="this is the cite of the day!"
    android:textSize="10sp"
    app:layout_constraintHorizontal_bias="0.514"
    app:layout_constraintLeft_toRightOf="@+id/buttonSaveTheBest"
    app:layout_constraintRight_toRightOf="parent"
    android:layout_marginTop="8dp"
    app:layout_constraintTop_toBottomOf="@+id/buttonGetCite"
    app:layout_constraintBottom_toBottomOf="parent"
    android:layout_marginBottom="8dp" />
</android.support.constraint.ConstraintLayout>

```

ActivityParser.java

```

package com.example.a1.appwithserviceandparser;

import android.content.Context;
import android.content.Intent;
import android.net.Uri;
import android.os.AsyncTask;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;

import org.json.JSONException;
import org.json.JSONObject;
import org.jsoup.Jsoup;
import org.jsoup.nodes.Document;
import org.jsoup.nodes.Element;
import org.jsoup.select.Elements;

import java.io.IOException;
import java.io.OutputStreamWriter;

public class ActivityParser extends AppCompatActivity {

    public String FILENAME = "cites.json";

    public Elements cite;
    public Elements author;
    public String citeOfTheDay = "";
    public String authorOfTheDay;
    public Button buttonGetCite;
    public TextView textCite;
    public TextView textAuthor;

    public Button buttonSite;
    public Button buttonSaveTheBest;
    public Button buttonCiteOfTheDay;

```

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_parser);

    buttonGetCite = (Button) findViewById(R.id.buttonGetCite);
    buttonSite = (Button) findViewById(R.id.buttonSite);
    buttonSaveTheBest = (Button) findViewById(R.id.buttonSaveTheBest);
    buttonCiteOfTheDay = (Button) findViewById(R.id.buttonCiteOfTheDay);
    textCite = (TextView) findViewById(R.id.textViewCiteMain);
    textAuthor = (TextView) findViewById(R.id.textViewAuthor);

    authorOfTheDay = "";
    if (citeOfTheDay == "") {
        new CiteThread().execute();
    }
    buttonGetCite.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            new CiteThread().execute();
        }
    });

    buttonSite.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Uri link = Uri.parse("https://citaty.info/random");
            Intent intent = new Intent(Intent.ACTION_VIEW, link);
            startActivity(intent);
        }
    });

    buttonSaveTheBest.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            String jstr = "";
            JSONObject jsonObj = new JSONObject();
            try {
                jsonObj.put("cite", citeOfTheDay);
                jsonObj.put("author", authorOfTheDay);
                jstr = jsonObj.toString();
            } catch (JSONException e) {
                e.printStackTrace();
            }
            try {
                OutputStreamWriter outputStreamWriter = new
                OutputStreamWriter(openFileOutput("cites.json", Context.MODE_PRIVATE));
                outputStreamWriter.write(jstr);
                outputStreamWriter.close();
            }
            catch (IOException e) {
                Log.e("Exception", "File write failed: " + e.toString());
            }
        }
    });

    buttonCiteOfTheDay.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent intent = new Intent(ActivityParser.this,
            MainActivity.class);
            intent.putExtra("citeOfTheDay", citeOfTheDay);
            intent.putExtra("authorOfTheDay", authorOfTheDay);

```

```

        startActivity(intent);
    }
    });
}

public class CiteThread extends AsyncTask<String, Void, String> {

    @Override
    protected String doInBackground(String... arg) {
        citeOfTheDay = "";
        authorOfTheDay = "";
        Document doc;
        try {
            doc = Jsoup.connect("https://citaty.info/random").get();
            cite = doc.select("#random-quote .node__content>.field.field-name-body.field-type-text-with-summary");
            for (Element e : cite) {
                citeOfTheDay = e.text();
                if (citeOfTheDay.equals("")) {
                    break;
                }
            }

            author = doc.select("#random-quote .node__content .field-type-taxonomy-term-reference a[title]");
            for (Element e : author) {
                String title = "";
                title = e.attr("title");
                if (title.equals("от имени автора")) {
                    continue;
                }
                if ((!title.equals("Автор цитаты"))&&(!title.equals("Цитата из книги"))&&(!title.equals("")))
                    authorOfTheDay = authorOfTheDay+title+":\n";
                authorOfTheDay = authorOfTheDay+e.text()+"\n";
            }

        } catch (IOException e) {
            e.printStackTrace();
        }
        return null;
    }

    @Override
    protected void onPostExecute(String result) {

        textCite.setText(citeOfTheDay);
        textAuthor.setText(authorOfTheDay);

    }
}

```

activity_service.xml

Activity лучшей цитаты. Получает из файла сохранённую лучшую цитату. По нажатию клавиши назначает эту цитату цитатой дня, передаёт её в главную activity и осуществляет переход на главную activity аналогично activity собственной цитаты пользователя.

```

<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.a1.appwithserviceandparser.ActivityService">

    <TextView
        android:id="@+id/textViewCiteMain"
        android:layout_width="347dp"
        android:layout_height="wrap_content"
        android:layout_marginLeft="8dp"
        android:layout_marginRight="8dp"
        android:layout_marginTop="16dp"
        android:textSize="18sp"
        app:layout_constraintHorizontal_bias="0.523"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/textViewAuthor1"
        android:layout_width="353dp"
        android:layout_height="wrap_content"
        android:layout_marginLeft="8dp"
        android:layout_marginTop="15dp"
        android:textAlignment="viewStart"
        android:textColor="@android:color/holo_blue_dark"
        android:textColorLink="@color/colorAccent"
        app:layout_constraintLeft_toLeftOf="parent"
        android:layout_marginRight="8dp"
        app:layout_constraintTop_toBottomOf="@+id/textViewCiteMain"
        app:layout_constraintRight_toRightOf="parent" />

    <Button
        android:id="@+id/buttonMakeCiteOfTheDay"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="8dp"
        android:layout_marginLeft="8dp"
        android:layout_marginRight="8dp"
        android:text="make it the cite of the day"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent" />

</android.support.constraint.ConstraintLayout>

```

ActivityService.java

```

package com.example.a1.appwithserviceandparser;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.content.Intent;
import android.widget.TextView;
import org.json.JSONException;

```



```

import org.json.JSONObject;
import java.io.BufferedReader;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;

public class ActivityService extends AppCompatActivity {

    public String FILENAME = "cites.json";

    public String cite;
    public String author;
    public TextView textCite;
    public TextView textAuthor;
    public Button buttonMakeCiteOfTheDay;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_service);

        textCite = (TextView) findViewById(R.id.textViewCiteMain);
        textAuthor = (TextView) findViewById(R.id.textViewAuthor1);
        //button = (Button) findViewById(R.id.buttonLast);
        buttonMakeCiteOfTheDay = (Button)
findViewById(R.id.buttonMakeCiteOfTheDay);
        String ret = "";

        try {
            InputStream inputStream = openFileInput(FILENAME);

            if ( inputStream != null ) {
                InputStreamReader inputStreamReader = new
InputStreamReader(inputStream);
                BufferedReader bufferedReader = new
BufferedReader(inputStreamReader);
                String receiveString = "";
                StringBuilder stringBuilder = new StringBuilder();

                while ( (receiveString = bufferedReader.readLine()) != null ) {
                    stringBuilder.append(receiveString);
                }

                inputStream.close();
                ret = stringBuilder.toString();
            }
        }
        catch (FileNotFoundException e) {
            Log.e("log1", "File not found: " + e.toString());
        } catch (IOException e) {
            Log.e("log2", "Can not read file: " + e.toString());
        }
        try {
            JSONObject jobj = new JSONObject(ret);
            cite = jobj.getString("cite");
            author = jobj.getString("author");
            textCite.setText(cite);
            textAuthor.setText(author);
        } catch (JSONException e) {
            e.printStackTrace();
        }
    }
}

```

```
        buttonMakeCiteOfTheDay.setOnClickListener(new View.OnClickListener() {  
            @Override  
            public void onClick(View v) {  
                Intent intent = new Intent(ActivityService.this,  
MainActivity.class);  
                intent.putExtra("citeOfTheDay", textCite.getText());  
                intent.putExtra("authorOfTheDay", textAuthor.getText());  
                startActivity(intent);  
            }  
        });  
    }  
}
```