## Хранение данных в общих настройках

### Цель работы

Научиться сохранять данные и состояние приложения в файле общих настроек, освоить операции чтения и записи в них, восстанавливать состояние из сохраненного файла

### Задания для выполнения

Создайте приложение, единственная активность которого состоит из формы, в которую пользователь вводит следующую информацию о студенте: имя, группа (выбор одного элемента из нескольких заранее определенных), возраст (в виде ползунка), дата рождения (в виде поля ввода даты).

### Методические указания

Для примера разберем более простое приложение, состоящее из одного текстового поля ввода. Для начала создадим макет главной активности:

<EditText

android:id="@+id/etName"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:ems="10"

android:inputType="textPersonName"

android:text=""

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintLeft\_toLeftOf="parent"

app:layout\_constraintRight\_toRightOf="parent"

app:layout\_constraintTop\_toTopOf="parent"/>

<Button

android:id="@+id/btnSave"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Сохранить"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintHorizontal\_bias="0.174"

app:layout\_constraintLeft\_toLeftOf="parent"

app:layout\_constraintRight\_toRightOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintVertical\_bias="0.717" />

<Button

android:id="@+id/btnLoad"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Восстановить"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintHorizontal\_bias="0.937"

app:layout\_constraintLeft\_toLeftOf="parent"

app:layout\_constraintRight\_toRightOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintVertical\_bias="0.717" />

Теперь перейдем к коду. Для начала нам понадобится создать три поля класса, содержащие наши элементы управления, так как мы будем обращаться с ним в нескольких методах:

Button btnSave, btnLoad;

EditText etName;

Содержание метода onCreate будет полностью стандартным: привязка к макету, получение ссылок на визуальные элементы и задание обработчиков событий:

**btnSave** = findViewById(R.id.***btnSave***);

**btnLoad** = findViewById(R.id.***btnLoad***);

**etName** = findViewById(R.id.***etName***);

**btnSave**.setOnClickListener(**this**);

**btnLoad**.setOnClickListener(**this**);

Теперь нам осталось создать метод onClick. Автозаполнение студии поможет вам в этом и сделает все за вас автоматически. Как обычно, когда один обработчик события привязан к нескольким элементам, нам понадобится множественный выбор:

@Override

**public void** onClick(View v) {

String name = **""**;

**switch** (v.getId()){

**case** R.id.***btnSave***:

**break**;

**case** R.id.***btnLoad***:

**break**;

**default**:

**break**;

}

}

Теперь, когда весь вспомогательный код готов, можно приступить к содержательной части - работе с файлом настроек. Для начала заполним ветку сохранения данных. Первым делом нам надо получить ссылку на файл настроек и оттуда получить объект редактора этого файла. Это можно сделать в две строки:

SharedPreferences pref = getPreferences(MODE\_PRIVATE);

SharedPreferences.Editor ed = pref.edit();

А можно и в одну, для краткости:

SharedPreferences.Editor ed = getPreferences(MODE\_PRIVATE).edit();

Теперь нам осталось только получить данные из поля ввода и записать их в файл. В файле настроек данные хранятся в виде набора пар имя-значение. Поэтому для записи нам нужно придумать уникальное имя (или ключ) для нашей записи. Это будет первым параметром в методе putString, который и записывает данные в файл настроек. Обратите внимание на команду коммита в редакторе:

name = **etName**.getText().toString();

ed.putString(**"name"**, name);

ed.commit();

Log.*i*(**"SPREF"**, name);

Еще проще будет заполнение метода чтения данных из файла. Нам всего лишь потребуется опять получить ссылку на файл (получать редактор уже не нужно, ведь мы будем только читать), получить оттуда строку по ключу и записать ее в наше текстовое поле:

SharedPreferences pref = getPreferences(***MODE\_PRIVATE***);

name = pref.getString(**"name"**, **""**);

**etName**.setText(name);

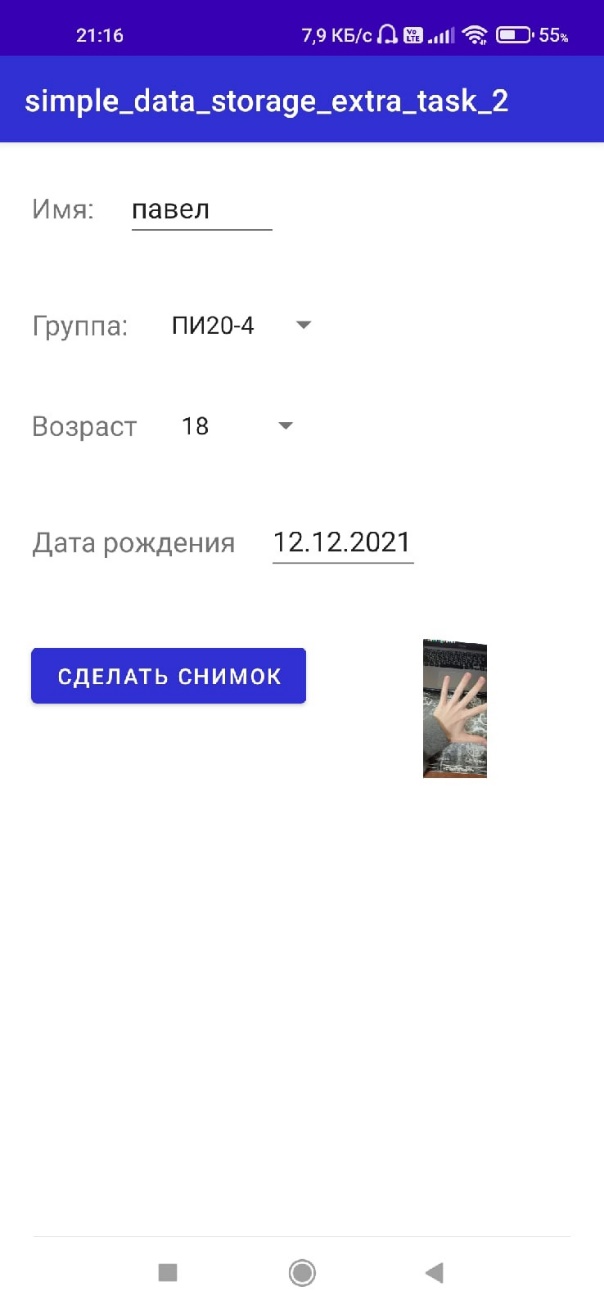
Основная работа уже сделана. Теперь можно заняться рефакторингом кода. Предоставляем вам самостоятельно выделить всю сутевую часть из обработчиков событий в отдельные функции, добавить другие запоминаемые поля ввода и проверить работоспособность программы.

### Контрольные вопросы

1. Что такое общие настройки и зачем они нужны?
2. Чем отличаются приватные и публичные общие настройки?
3. Какие данные не целесообразно хранить в файле общих настроек?

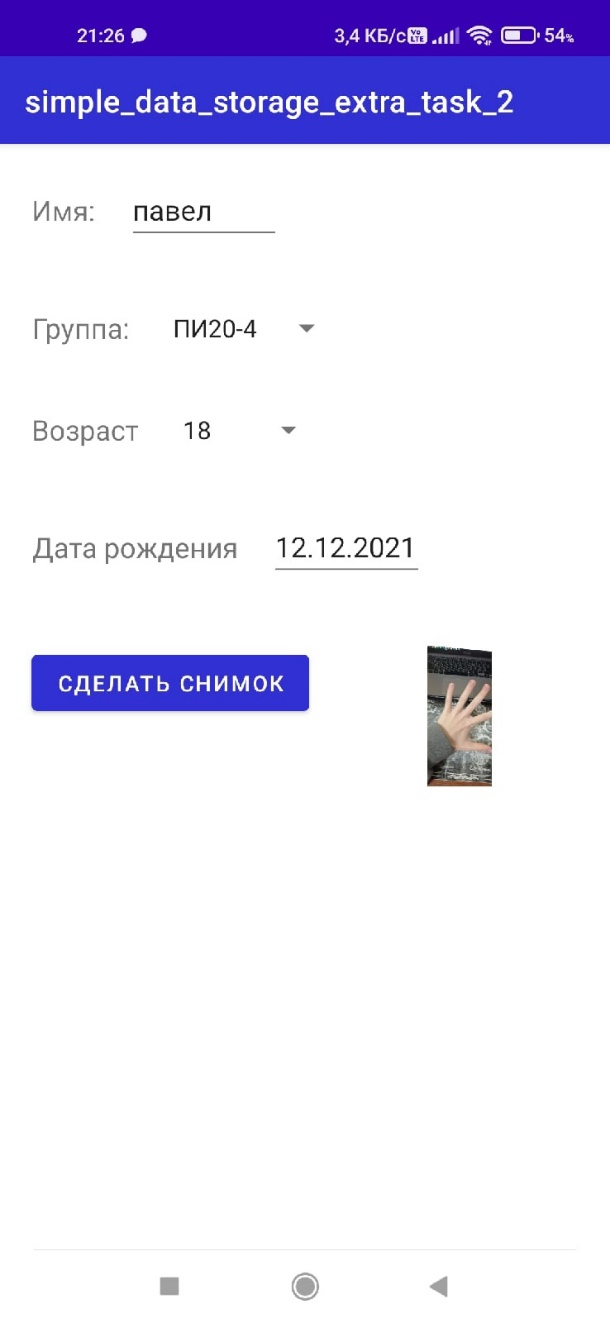
### Дополнительные задания

1. Модифицируйте программу таким образом, чтобы все необходимые данные автоматически сохранялись при остановке приложения и восстанавливались при возобновлении работы.
2. (\*) Добавьте возможность в форме делать снимок в камеры телефона. Загруженный снимок должен сохраняться в отдельный файл приложения и так же восстанавливаться при следующем запуске.



После перезагрузки:

Изображение выглядит как текст, электроника

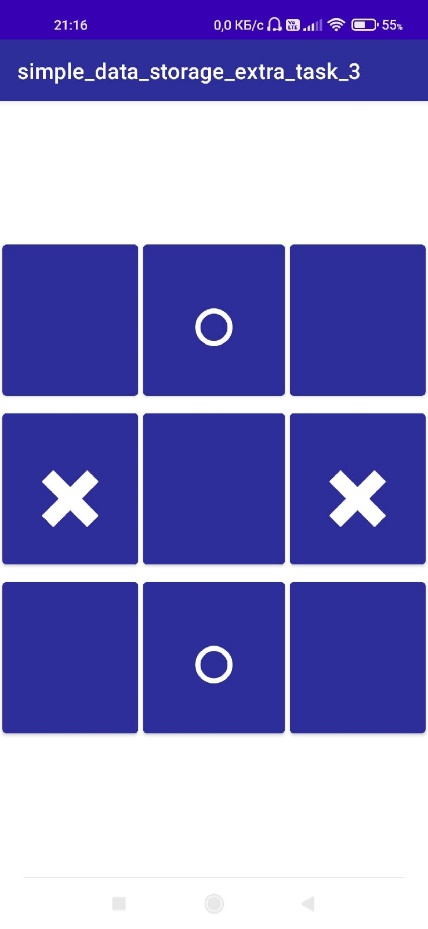
Автоматически созданное описаниеMainActivity.java:

package *com.example.simple\_data\_storage\_extra\_task\_2*;  
  
import *androidx.annotation.Nullable*;  
import *androidx.appcompat.app.AppCompatActivity*;  
  
import *android.annotation.SuppressLint*;  
import *android.content.*Context;  
import *android.content.ContextWrapper*;  
import *android.content.Intent*;  
import *android.content.SharedPreferences*;  
import *android.graphics.Bitmap*;  
import *android.graphics.BitmapFactory*;  
import *android.os.Bundle*;  
import *android.provider.MediaStore*;  
import *android.view.View*;  
import *android.widget.*AdapterView;  
import *android.widget.ArrayAdapter*;  
import *android.widget.EditText*;  
import *android.widget.ImageView*;  
import *android.widget.Spinner*;  
  
import *java.io.File*;  
import *java.io.FileInputStream*;  
import *java.io.FileNotFoundException*;  
import *java.io.FileOutputStream*;  
import *java.io.IOException*;  
  
public class *MainActivity* extends *AppCompatActivity* implements *View*.*OnClickListener*, AdapterView.*OnItemSelectedListener* {  
 final *String* DATA = "data";  
 final *String* NAME = "name";  
 final *String* GROUP = "group";  
 final *String* AGE = "age";  
 final *String* DATE = "date";  
 final *String* IMAGE\_PATH = "image";  
 final *String* IMAGE\_NAME = "image.jpeg";  
  
 *EditText* etName, etDate;  
 *Spinner* spGroup, spAge;  
 *ArrayAdapter*<*String*> aaGroup;  
 *ArrayAdapter*<*String*> aaAge;  
 *String*[] groups = {"ПИ20-4", "ПИ20-5", "ПИ20-6"};  
 *String*[] ages = new String[82];  
 int currentGroupId, ageId;  
 *ImageView* ivImage;  
 *Bitmap* bmImage;  
  
 *@Override* protected void onCreate(*Bundle savedInstanceState*) {  
 super.onCreate(*savedInstanceState*);  
 setContentView(*R*.*layout*.activity\_main);  
  
 etName = findViewById(*R*.*id*.etName);  
 etDate = findViewById(*R*.*id*.etDate);  
  
 spGroup = findViewById(*R*.*id*.spGroup);  
 spAge = findViewById(*R*.*id*.spAge);  
  
  
 for (int i = 0; i < ages.length; i++) {  
 ages[i] = *String*.*valueOf*(i + 18);  
 }  
  
 aaGroup = new ArrayAdapter<>(this, *android.R*.*layout*.simple\_spinner\_item, groups);  
 aaAge = new ArrayAdapter<>(this, *android.R*.*layout*.simple\_spinner\_item, ages);  
  
 aaGroup.setDropDownViewResource(*android.R*.*layout*.simple\_spinner\_dropdown\_item);  
 aaAge.setDropDownViewResource(*android.R*.*layout*.simple\_spinner\_dropdown\_item);  
  
 spGroup.setAdapter(aaGroup);  
 spAge.setAdapter(aaAge);  
  
 spGroup.setSelection(0);  
 spAge.setSelection(0);  
  
 spGroup.setOnItemSelectedListener(this);  
 spAge.setOnItemSelectedListener(this);  
  
 findViewById(*R*.*id*.btnImage).setOnClickListener(this);  
  
 ivImage = findViewById(*R*.*id*.ivImage);  
  
 *SharedPreferences* preferences = getPreferences(MODE\_PRIVATE);  
  
 etName.setText(preferences.getString(NAME, ""));  
 spGroup.setSelection(preferences.getInt(GROUP, 0));  
 spAge.setSelection(preferences.getInt(AGE, 0));  
 etDate.setText(preferences.getString(DATE, ""));  
 loadImage(preferences.getString(IMAGE\_PATH, ""));  
 }  
  
 *@Override* protected void onPause() {  
 super.onPause();  
 *SharedPreferences*.*Editor* preferences = getPreferences(MODE\_PRIVATE).edit();  
 preferences.putString(NAME, etName.getText().toString());  
 preferences.putInt(GROUP, currentGroupId);  
 preferences.putInt(AGE, ageId);  
 preferences.putString(DATE, etDate.getText().toString());  
 preferences.putString(IMAGE\_PATH, saveImage(bmImage));  
 preferences.apply();  
 }  
  
 *@SuppressLint*("NonConstantResourceId")  
 *@Override* public void onClick(*View view*) {  
 if (*view*.getId() == *R*.*id*.btnImage) {  
 *Intent* intent = new Intent(*MediaStore*.ACTION\_IMAGE\_CAPTURE);  
 startActivityForResult(intent, 1);  
 }  
 }  
  
 *@Override* protected void onActivityResult(int *requestCode*, int *resultCode*, *@Nullable Intent data*) {  
 super.onActivityResult(*requestCode*, *resultCode*, *data*);  
 if (*resultCode* == RESULT\_OK) {  
 if (*data* != null) {  
 *Bundle* bundle = *data*.getExtras();  
 bmImage = (*Bitmap*) bundle.get(DATA);  
 ivImage.setImageBitmap(bmImage);  
 }  
 }  
 }  
  
 *@SuppressLint*("NonConstantResourceId")  
 *@Override* public void onItemSelected(AdapterView<?> *adapterView*, *View view*, int *i*, long *l*) {  
 switch (*view*.getId()) {  
 case *R*.*id*.spGroup:  
 currentGroupId = *i*;  
 break;  
 case *R*.*id*.spAge:  
 ageId = *i*;  
 break;  
 default:  
 break;  
 }  
 }  
  
 *@Override* public void onNothingSelected(AdapterView<?> *adapterView*) {  
 }  
  
 private void loadImage(*String path*) {  
 try {  
 bmImage = *BitmapFactory*.*decodeStream*(new FileInputStream(new File(*path*, IMAGE\_NAME)));  
 ivImage.setImageBitmap(bmImage);  
 } catch (*FileNotFoundException e*) {  
 *e*.printStackTrace();  
 }  
 }  
  
 private *String* saveImage(*Bitmap bitmap*) {  
 *ContextWrapper* contextWrapper = new ContextWrapper(getApplicationContext());  
 *File* directory = contextWrapper.getDir("imageDir", Context.MODE\_PRIVATE);  
 *File* path = new File(directory, IMAGE\_NAME);  
 *FileOutputStream* fileOutputStream = null;  
 try {  
 fileOutputStream = new FileOutputStream(path);  
 *bitmap*.compress(*Bitmap*.*CompressFormat*.JPEG, 100, fileOutputStream);  
 } catch (*Exception e*) {  
 *e*.printStackTrace();  
 } finally {  
 try {  
 if (fileOutputStream != null) {  
 fileOutputStream.close();  
 }  
 } catch (*IOException e*) {  
 *e*.printStackTrace();  
 }  
 }  
 return directory.getAbsolutePath();  
 }  
}

activity\_main.xml:

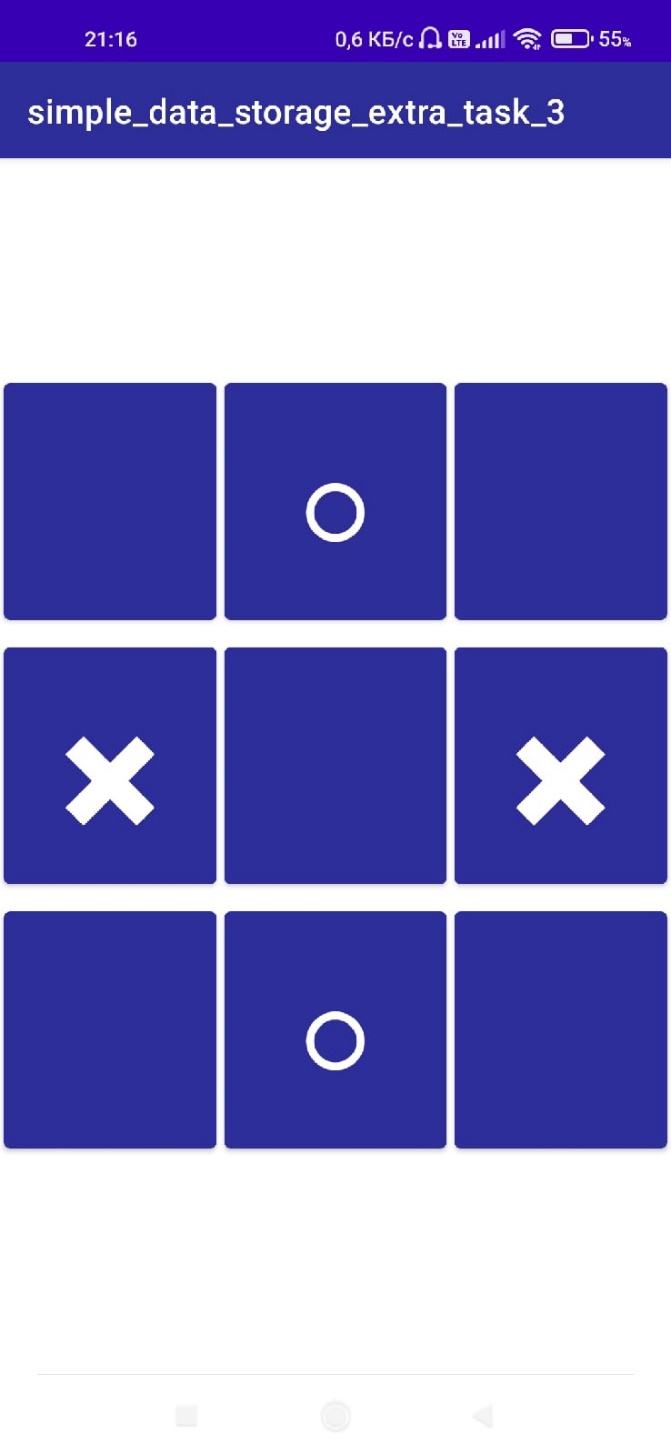
<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 tools:context=".MainActivity">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="10dp"  
 android:orientation="horizontal">  
  
 <TextView  
 android:id="@+id/tvName"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="match\_parent"  
 android:layout\_margin="10dp"  
 android:gravity="center"  
 android:labelFor="@+id/etName"  
 android:text="@string/name"  
 android:textSize="18sp" />  
  
 <EditText  
 android:id="@+id/etName"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="match\_parent"  
 android:layout\_margin="10dp"  
 android:autofillHints="Name"  
 android:inputType="text"  
 android:minWidth="100dp" />  
 </LinearLayout>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="10dp"  
 android:orientation="horizontal">  
  
 <TextView  
 android:id="@+id/tvGroup"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="match\_parent"  
 android:layout\_margin="10dp"  
 android:gravity="center"  
 android:labelFor="@+id/spGroup"  
 android:text="@string/group"  
 android:textSize="18sp" />  
  
 <Spinner  
 android:id="@+id/spGroup"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="match\_parent"  
 android:layout\_margin="10dp"  
 android:minWidth="100dp" />  
 </LinearLayout>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="10dp"  
 android:orientation="horizontal">  
  
 <TextView  
 android:id="@+id/tvAge"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="match\_parent"  
 android:layout\_margin="10dp"  
 android:gravity="center"  
 android:labelFor="@+id/spAge"  
 android:text="@string/age"  
 android:textSize="18sp" />  
  
 <Spinner  
 android:id="@+id/spAge"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="match\_parent"  
 android:layout\_margin="10dp"  
 android:minWidth="100dp" />  
 </LinearLayout>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="10dp"  
 android:orientation="horizontal">  
  
 <TextView  
 android:id="@+id/tvDate"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="match\_parent"  
 android:layout\_margin="10dp"  
 android:gravity="center"  
 android:labelFor="@+id/etDate"  
 android:text="@string/date"  
 android:textSize="18sp" />  
  
 <EditText  
 android:id="@+id/etDate"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="match\_parent"  
 android:layout\_margin="10dp"  
 android:autofillHints="Name"  
 android:inputType="date"  
 android:minWidth="100dp" />  
 </LinearLayout>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="10dp"  
 android:orientation="horizontal">  
  
 <Button  
 android:id="@+id/btnImage"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="10dp"  
 android:text="@string/makePhoto" />  
  
 <ImageView  
 android:id="@+id/ivImage"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="10dp"  
 android:contentDescription="@string/userPhoto" />  
 </LinearLayout>  
  
</LinearLayout>

1. (\*) Создайте мобильную игру в крестики-нолики используя специально подготовленные графические элементы. Модифицируйте программу таким образом, чтобы прогресс игрока сохранялся между запусками программы. Придумайте схему хранения состояния игры.



После перезагрузки:

Изображение выглядит как текст, электроника

Автоматически созданное описание

MainActivity.java:

package *com.example.simple\_data\_storage\_extra\_task\_3*;  
  
import *android.annotation.SuppressLint*;  
import *android.content.SharedPreferences*;  
import *android.os.Bundle*;  
import *android.view.View*;  
import *android.widget.Button*;  
import *android.widget.Toast*;  
  
import *androidx.appcompat.app.AppCompatActivity*;  
  
public class *MainActivity* extends *AppCompatActivity* implements *View*.*OnClickListener* {  
 final *String* CROSS = "✖";  
 final *String* ZERO = "⭘";  
 final *String* EMPTY = "";  
 final *String* BTN11 = "btn11", BTN12 = "btn12", BTN13 = "btn13";  
 final *String* BTN21 = "btn21", BTN22 = "btn22", BTN23 = "btn23";  
 final *String* BTN31 = "btn31", BTN32 = "btn32", BTN33 = "btn33";  
 final *String* MOVES = "moves";  
 final int CELLS = 9;  
 *String*[] BUTTONS = {BTN11, BTN12, BTN13, BTN21, BTN22, BTN23, BTN31, BTN32, BTN33};  
 *Button* btn11, btn12, btn13;  
 *Button* btn21, btn22, btn23;  
 *Button* btn31, btn32, btn33;  
 *Button*[] buttons;  
 int moves = 0;  
  
 *@Override* protected void onCreate(*Bundle savedInstanceState*) {  
 super.onCreate(*savedInstanceState*);  
 setContentView(*R*.*layout*.activity\_main);  
  
 btn11 = findViewById(*R*.*id*.btn11);  
 btn12 = findViewById(*R*.*id*.btn12);  
 btn13 = findViewById(*R*.*id*.btn13);  
 btn21 = findViewById(*R*.*id*.btn21);  
 btn22 = findViewById(*R*.*id*.btn22);  
 btn23 = findViewById(*R*.*id*.btn23);  
 btn31 = findViewById(*R*.*id*.btn31);  
 btn32 = findViewById(*R*.*id*.btn32);  
 btn33 = findViewById(*R*.*id*.btn33);  
  
 buttons = new Button[]{btn11, btn12, btn13, btn21, btn22, btn23, btn31, btn32, btn33};  
  
 for (*Button* button : buttons) {  
 button.setOnClickListener(this);  
 }  
  
 clear();  
  
 *SharedPreferences* preferences = getPreferences(MODE\_PRIVATE);  
 for (int i = 0; i < CELLS; i++) {  
 buttons[i].setText(preferences.getString(BUTTONS[i], ""));  
 }  
 moves = preferences.getInt(MOVES, 0);  
 }  
  
 *@Override* protected void onPause() {  
 super.onPause();  
 *SharedPreferences*.*Editor* preferences = getPreferences(MODE\_PRIVATE).edit();  
 for (int i = 0; i < CELLS; i++) {  
 preferences.putString(BUTTONS[i], buttons[i].getText().toString());  
 }  
 preferences.putInt(MOVES, moves);  
 preferences.apply();  
 }  
  
 public void check(*Button first*, *Button second*, *Button third*) {  
 if (*first*.getText().toString().equals(*second*.getText().toString()) && *second*.getText().toString().equals(*third*.getText().toString()) && !*first*.getText().toString().equals(EMPTY)) {  
 *Toast*.*makeText*(getApplicationContext(), "Игрок " + *first*.getText().toString() + " одержал победу!", *Toast*.LENGTH\_LONG).show();  
 clear();  
 } else if (moves == 9) {  
 *Toast*.*makeText*(getApplicationContext(), "Ничья!", *Toast*.LENGTH\_LONG).show();  
 clear();  
 }  
 }  
  
 public void setChar(*Button button*) {  
 if (*button*.getText().toString().equals(EMPTY)) {  
 if (moves % 2 == 0) {  
 *button*.setText(CROSS);  
 } else {  
 *button*.setText(ZERO);  
 }  
 moves++;  
 }  
 }  
  
 public void clear() {  
 for (*Button* button : buttons) {  
 button.setText(EMPTY);  
 }  
 moves = 0;  
 }  
  
 *@SuppressLint*("NonConstantResourceId")  
 *@Override* public void onClick(*View view*) {  
 switch (*view*.getId()) {  
 case *R*.*id*.btn11:  
 setChar(btn11);  
 check(btn11, btn12, btn13);  
 check(btn11, btn22, btn33);  
 check(btn11, btn21, btn31);  
 break;  
 case *R*.*id*.btn12:  
 setChar(btn12);  
 check(btn11, btn12, btn13);  
 check(btn12, btn22, btn32);  
 break;  
 case *R*.*id*.btn13:  
 setChar(btn13);  
 check(btn11, btn12, btn13);  
 check(btn13, btn22, btn31);  
 check(btn13, btn23, btn33);  
 break;  
 case *R*.*id*.btn21:  
 setChar(btn21);  
 check(btn11, btn21, btn31);  
 check(btn21, btn22, btn23);  
 break;  
 case *R*.*id*.btn22:  
 setChar(btn22);  
 check(btn11, btn22, btn33);  
 check(btn31, btn22, btn13);  
 check(btn21, btn22, btn23);  
 check(btn12, btn22, btn32);  
 break;  
 case *R*.*id*.btn23:  
 setChar(btn23);  
 check(btn13, btn23, btn33);  
 check(btn21, btn22, btn23);  
 break;  
 case *R*.*id*.btn31:  
 setChar(btn31);  
 check(btn11, btn21, btn31);  
 check(btn13, btn22, btn31);  
 check(btn31, btn32, btn32);  
 break;  
 case *R*.*id*.btn32:  
 setChar(btn32);  
 check(btn31, btn32, btn33);  
 check(btn12, btn22, btn32);  
 break;  
 case *R*.*id*.btn33:  
 setChar(btn33);  
 check(btn31, btn32, btn33);  
 check(btn11, btn22, btn33);  
 check(btn13, btn23, btn33);  
 break;  
 }  
 }  
}

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">  
  
 <Button  
 android:id="@+id/btn12"  
 style="@style/CustomStyle"  
 android:layout\_above="@+id/btn22"  
 android:layout\_centerHorizontal="true" />  
  
 <Button  
 android:id="@+id/btn13"  
 style="@style/CustomStyle"  
 android:layout\_above="@id/btn23"  
 android:layout\_toEndOf="@id/btn12" />  
  
 <Button  
 android:id="@+id/btn11"  
 style="@style/CustomStyle"  
 android:layout\_above="@id/btn21"  
 android:layout\_toStartOf="@+id/btn22" />  
  
 <Button  
 android:id="@+id/btn22"  
 style="@style/CustomStyle"  
 android:layout\_centerInParent="true" />  
  
 <Button  
 android:id="@+id/btn23"  
 style="@style/CustomStyle"  
 android:layout\_centerVertical="true"  
 android:layout\_toEndOf="@+id/btn22" />  
  
 <Button  
 android:id="@+id/btn21"  
 style="@style/CustomStyle"  
 android:layout\_centerVertical="true"  
 android:layout\_toStartOf="@+id/btn22" />  
  
 <Button  
 android:id="@+id/btn32"  
 style="@style/CustomStyle"  
 android:layout\_below="@id/btn22"  
 android:layout\_centerHorizontal="true" />  
  
 <Button  
 android:id="@+id/btn33"  
 style="@style/CustomStyle"  
 android:layout\_below="@id/btn23"  
 android:layout\_toEndOf="@+id/btn32" />  
  
 <Button  
 android:id="@+id/btn31"  
 style="@style/CustomStyle"  
 android:layout\_below="@id/btn21"  
 android:layout\_toStartOf="@+id/btn22" />  
</RelativeLayout>

custom\_style.xml:

<?xml version="1.0" encoding="utf-8"?>  
<resources>  
  
 <style name="CustomStyle">  
 <item name="android:layout\_margin">2dp</item>  
 <item name="android:layout\_width">130dp</item>  
 <item name="android:layout\_height">150dp</item>  
 <item name="android:gravity">center</item>  
 <item name="android:textColor">@color/white</item>  
 <item name="android:textSize">70sp</item>  
 </style>  
</resources>