Лекция #22. Parcelable

Возможность сериализации объектов предоставляется напрямую инфраструктурой языка **Java**. Однако **Android** также предоставляет интерфейс **Parcelable**, который по сути также позволяет сериализовать объекты, как и Serializable, но является более оптимизированным для Android. И подобные объекты **Parcelable** также можно передавать между двумя **activity** или использовать каким-то иным образом.

Например, в прошлой теме данные передавались между **activity** в виде объектов **User**, которые использовали сериализацию. Теперь пусть класс **User** применяет интерфейс **Parcelable**:

```
package com.awkitsune.activitiesexample

import android.os.Parcel
import android.os.Parcelable

class User() : Parcelable {
    private var name: String = ""
    private var company: String = ""
    private var age: Int = 0

    constructor(parcel: Parcel) : this() {
        name = parcel.readString().toString()
        company = parcel.readString().toString()
        age = parcel.readInt()
    }
    companion object CREATOR : Parcelable.Creator<User> {
        override fun createFromParcel(parcel: Parcel): User {
            return User(parcel)
        }

        override fun newArray(size: Int): Array<User?> {
            return arrayOfNulls(size)
```

```
        }
    }

    constructor(_name: String, _company: String, _age: Int) : this() {
        name = _name
        company = _company
        age = _age
    }

    fun setName(_name: String){
        name = _name
    }

    fun getName(): String {
        return name
    }

    fun setCompany(_company: String){
        company = _company
    }

    fun getCompany(): String {
        return company
    }

    fun setAge(_age: Int){
        age = _age
    }

    fun getAge(): Int {
        return age
    }

    override fun describeContents(): Int {
        return 0
    }

    override fun writeToParcel(dest: Parcel?, flags: Int) {
        dest?.writeString(name)
        dest?.writeString(company)
        dest?.writeInt(age)
    }
}
```

Интерфейс **android.os.Parcelable** предполагает реализацию двух методов: **describeContents()** и **writeToParcel(). Первый** метод описывает контент и возвращает некторое числовое значение. **Второй** метод пишет в объект **Parcel** содержимое объекта **User**.

Для записи данных объекта в Parcel используется ряд методов, каждый из которых предназначен для определенного типа данных. Основные методы:

- **writeString()**
- **writeInt()**
- **writeFloat()**
- **writeDouble()**
- **writeByte()**
- **writeLong()**
- **writeIntArray()**
- **writeValue()** (записывает объект типа **Object**)
- **writeParcelable()** (записывает объект типа **Parcelable**)

Кроме того, объект Parcelable должен содержать статическое поле **CREATOR**, которое представляет объект **Creator<User>**. Причем этот объект реализует два метода. Они нужны для создания их ранее сериализованных данных исходных объектов типа **User**.

Так, метод **newArray()** создает массив объект **User**.

Метод **createFromParcel** создает из **Parcel** новый объект типа **User**. То есть этот метод противоположен по действию методу **writeToParcel**. Для получения данных из Parcel применяются методы типа **readString()**, **readInt()**, **readParcelable()** и так далее - для чтения определенных типов данных.

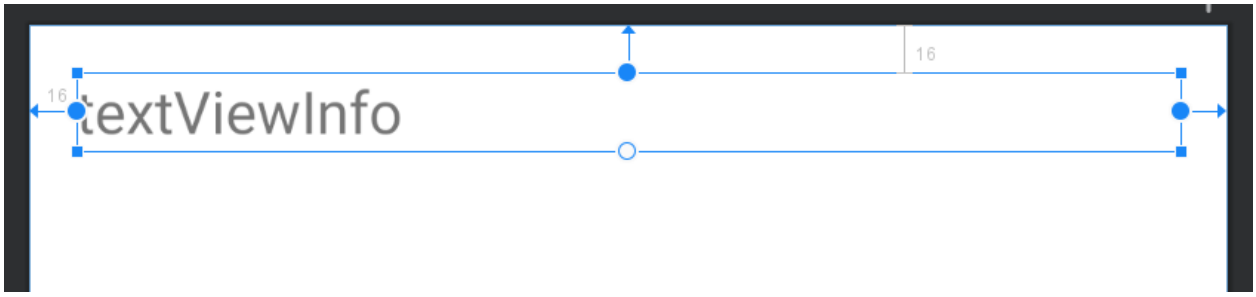
Причем важно, что данные в **createFromParcel** считываются из объекта **Parcel** именно в том порядке, в котором они добавляются в этот объект в методе **writeToParcel**.

Допустим в activity, которая называется **SecondActivity** мы будем получать объект **User**:

```
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_second)

    val textView: TextView = findViewById(R.id.textViewInfo)

    var user: User = User()

    val arguments: Bundle? = intent.extras
    if (arguments != null){
        user =
arguments.getParcelable<User>(user.javaClass.simpleName) as User

        textView.text = "Name: ${user.getName()} \nCompany:
```

```
${user.getCompany()} \nAge: ${user.getAge()}"
    }
}
```

Так же добавим **textViewInfo** в интерфейс **SecondActivity**:



Для получения объекта Parcelable, переданного в activity, применяется метод **getParcelable()**.

Для тестирования передачи Parcelable определим в файле **activity_main.xml** простейший интерфейс для MainActivity:

Здесь определены три текстовых поля для ввода данных и кнопка.

А в коде **MainActivity** определим передачу данных в SecondActivity:

```
fun buttonSendClick(view: View){
    val nameText: TextView = findViewById(R.id.editTextName)
    val ageText: TextView = findViewById(R.id.editTextAge)
    val companyText: TextView = findViewById(R.id.editTextCompany)

    val name = nameText.text.toString()
    val age = ageText.text.toString().toInt()
    val company = companyText.text.toString()

    val user: User = User(name, company, age)

    val intent: Intent = Intent(this@MainActivity,
SecondActivity::class.java)
    intent.putExtra(user.javaClass.simpleName, user)
```

```
    startActivity(intent)
}
```