

№3-amaliyot ishi

Satrlar va ranglar qiymatlaridan foydalanish

1. Ishdan maqsad

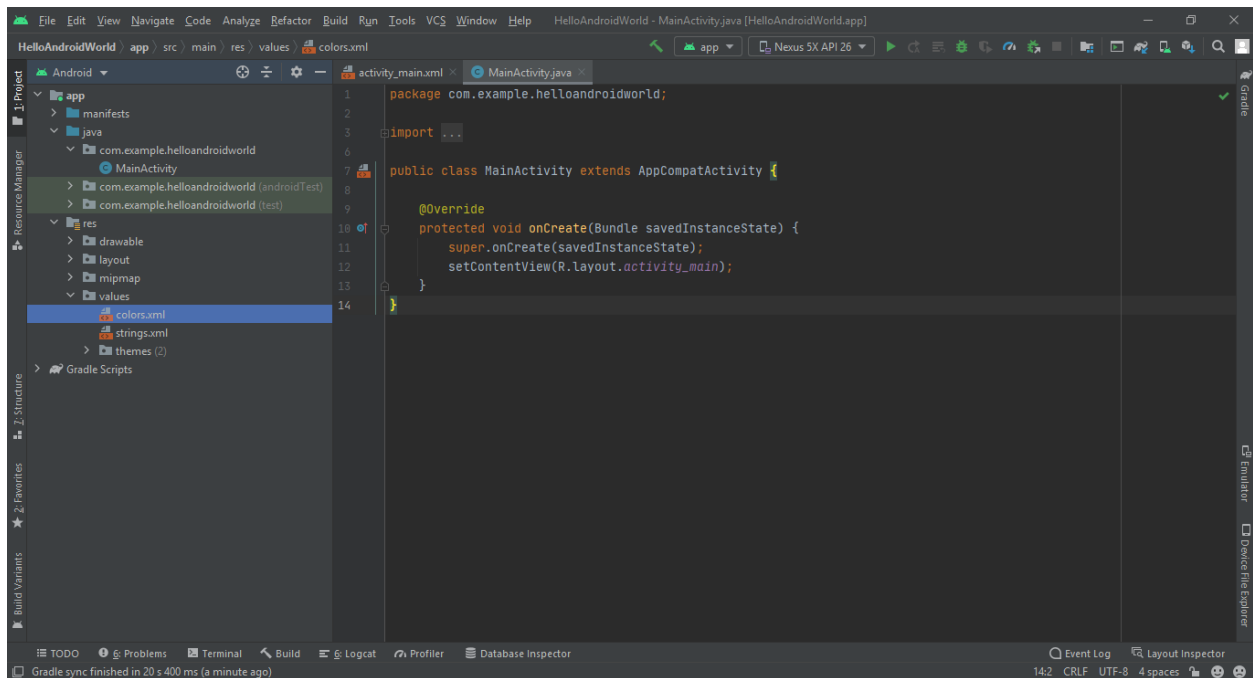
Amaliyot ishidan maqsad ilovaning tashqi ko‘rinishini o‘zgartirish uchun satr konstantalaridagi rang, shaffoflik qiymatlarini o‘zgartirish, HTML-teglar bilan ishlashni o‘rganish.

2. Topshiriq

1. *solor* faylini yaratish.
2. `res/values/colors.xml` tarkibini tahrir qiling.
3. `res/values/strings.xml` tarkibini tahrir qiling.
4. Satr konstantalaridagi rang, shaffoflik qiymatlarini o‘zgartirish, HTML-teglar bilan ishlab ko‘ring.

1. Amaliyot ishini bajarish bo‘yicha ko‘rsatmlar

1. Mavjud *Hello Android World* loyihada *colors* File → New → Android XML File faylini yarating agar mavjud bo‘lsa unga kiring:



3.1-rasm. *res/values* katalogda *colors* faylini yaratish

2. *res/values/colors.xml* fayli tarkibini tahrir qiling:

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<resources>
```

```
<color name="purple_200">#FFBB86FC</color>
```

```
<color name="purple_500">#FF6200EE</color>
```

```
<color name="purple_700">#FF3700B3</color>
```

```
<color name="teal_200">#FF03DAC5</color>
```

```
<color name="teal_700">#FF018786</color>
```

```
<color name="black">#FF000000</color>
```

```
<color name="white">#FFFFFFFF</color>
```

```
<color name="qizil">#FF0000</color>
```

```
<color name="sariq">#FFE500</color>
<color name="yashil">#59FF00</color>
</resources>
```

3. res/values/strings.xml fayli tarkibini tahrir qiling:

```
<resources>
  <string name="app_name">HelloAndroidWorld</string>
  <string name="hello">Hello World!</string>
</resources>
```

4. res/layout/main.xml belgilash fayliga o‘zgartirishlar kiriting:

```
<?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"
  xmlns:tools="http://schemas.android.com/tools"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:background="@color/yashil"
  tools:context=".MainActivity">

  <TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/hello"
```

```
android:textColor="@color/qizil"
android:background="@color/sariq"
android:textSize="34sp"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintRight_toRightOf="parent"
app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

5. Barcha saqlanmagan fayllarni saqlang va ilovani ishga tushiring. Ishga tushurishdan oldin Manifest file ga kirib **application** qatoriga **tools:ignore="AllowBackup"** va **activity** qatoriga **android:exported="true"** comandalarini qo'shib qo'ying. 2021 – yildan buyon AndroidManifest.xml file da shu buyruqlar bo'lishi talab qilinyabdi. Shundan so'ng ishga tushurishingiz mumkun:



3.2-rasm. Ilovani ishga tushirish

6. Ilovaning tashqi ko‘rinishi biz aniqlagan resurslarga muvofiq o‘zgarganligiga, juda yomon xunuk bo‘lib qolganligiga ishonch hosil qiling.

7. Ilovaning tashqi ko‘rinishini o‘zgartirish uchun satr konstantalaridagi rang, shaffoflik va HTML-teglar qiymatlarini o‘zgartirish bilan ishlab ko‘ring

O‘lchamlar

O'lchamlarga ko'rsatishlar ko'pincha resurslar ichida uslublar va belgilashlar bilan, masalan ramkaning qalinligi yoki shriftning kattaligini ko'rsatishda uchraydi o'lchamlarni tavsiflash uchun o'lchamlilik turli ko'rsatiladigan *<dimen>* tegi ishlatiladi:

- px – real ekran piksellari;
- in – fizik dyuymlar;
- pt – 1/72 dyuym, ekranning fizik o'lchamidan hisoblanadi;
- mm – fizik millimetrlar, ekranning fizik o'lchamidan hisoblanadi;
- dp – ekran zichligiga bog'liq bo'lmagan “mustaqil” piksellar, 160 dpi etalon zichligida bitta pikselga teng, dip sifatida ham ko'rsatilishi mumkin, ko'pincha ramkalar va maydonlar o'lchamlarini ko'rsatish uchun ishlatiladi;

- sp – masshtabga bog'liq bo'lmagan “mustaqil” piksellar, dpga o'xshash, lekin shriftlar o'lchamlarini foydalanuvchi sozlashini (yirik, mayda, o'rtacha) ham hisobga oladi, shuning uchun shriftlarni tavsiflash uchun ishlatiladi. “Katta” shrift va “satndart” ramkani tavsiflashga misol:

```
<dimen name="standard_border">5dp</dimen>
```

```
<dimen name="large_font_size">16sp</dimen>
```

Vizual uslublar va mavzular

Uslublar va mavzular Taqdim etishlarda (View) ishlatiladigan atributlar, ko'pincha bu ranglar va shriftlar yoramida ilovaning tashqi ko'rinishi birligini saqlashga imkon beradi.

Ilovaning tashqi ko'rinishi ilovalar Manifestida uslublarni (bezash mavzularini) o'zgartirishda oson o'zgartiriladi.

Uslubni yaratish uchun *name* atribut *<style>* teg ishlatiladi, u *<item>* elementlariga ega bo'lib, ulardan har biri, o'z navbatida, parametr turini

ko‘rsatadigan (masalan, rang yoki o‘lcham) *name* atributiga ega. *<item>* elementi ichida parametrning qiymati saqlanadi:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
<style name="StyleName">
<item name="attributeName">attributeValue</item>
[ ... еще элементы <item> ... ] </style>
</resources>
```

<style> tegda parent atributni ko‘rsatish mumkin, bu yangi uslubga mavjud uslubdan sezilarsiz farqlarni kiritish zaruratida uslublarni “olinishini” mumkin qiladi:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
<style name="NormalText">
<item name="android:textSize">14sp</item>
<item name="android:textColor">#111</item>
</style>
<style name="SmallText" parent="NormalText">
<item name="android:textSize">8sp</item>
</style>
</resources>
```

Tasvirlar

Drawable resurslari rastr tasvirlariga ega. Bu XML formatda tavsiflanadigan *LevelListDrawables* va *StateListDrawables* kabi murakkab tarkibiy resurslar, shuningdek *NinePatch* cho‘ziladigan rastr tasvirlari bo‘lishi mumkin. *Drawable* resurslari `res/drawable-*` kataloglarda alohida fayllar ko‘rinishida saqlanadi. Bunday resurslar uchun identifikatorlar bo‘lib pastki registrdagi nomlar (kengaytirishsiz) xizmat qiladi. PNG (afzalroq), JPEG va GIF formatlari qo‘llaniladi.

Belgilash

Belgilanadigan resurslardan (layout) foydalanish tufayli ishlab chiquvchi dasturning mantig‘ini uning tashqi ko‘rinishidan ajratish imkoniyatini beradi. XML format faylida aniqlangan belgilashni Aktivlikda bizning ilovamizdagi *onCreate* usulida ishlatilganidek *setContentView* usuli orqali ishlatish uchun yuklash mumkin:

```
setContentView(R.layout.main);
```

Belgilashli har bir resurs `res/layout` katalogidagi alohida faylda saqlanadi. Faylning nomi bu resursning identifikatori sifatida (odatdagidek kengaytirishsiz) ishlatiladi. Bizning o‘quv ilovamizni yaratishda yangi loyihalarni yaratish masteri biz uchun biz tahrir qilib bo‘lgan `res/layout/main.xml` faylini yaratdi:

```
<?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"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/yashil"
```



```
tools:context=".MainActivity">
```

```
<TextView
```

```
    android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content"
```

```
    android:text="@string/hello"
```

```
    android:textColor="@color/qizil"
```

```
    android:background="@color/sariq"
```

```
    android:textSize="34sp"
```

```
    app:layout_constraintBottom_toBottomOf="parent"
```

```
    app:layout_constraintLeft_toLeftOf="parent"
```

```
    app:layout_constraintRight_toRightOf="parent"
```

```
    app:layout_constraintTop_toTopOf="parent" />
```

```
</androidx.constraintlayout.widget.ConstraintLayout>
```

Bu fayl *androidx.constraintlayout.widget.ConstraintLayout* belgilashga ega bo‘lib, u *strings* resursda tavsiflangan *hello* nomli satrning tarkibini aks ettiradigan *TextView* elementi uchun konteyner hisoblanadi.

Animatsiya

Android ikkita turdagi animatsiyalarni qo‘llaydi:

- tasvirni berilgan davomiylikda ekranga ketma-ket chiqaradigan qadamlab animatsiya;
- oraliq kadrlarga mo‘ljallga asoslangan animatsiyalash, bu holda turli o‘zgartirishlar – aylantirish, cho‘zish, ko‘chirish va qorong‘ilashtirish qo‘llaniladi.

Bu barcha transformatsiyalar res/anim katalogidagi XML-faylda tavsiflanadi. Maqsadli element bir vaqtda 270 graduslarga buriladigan, siqiladigan va asta-sekin yo‘qoladigan animatsiyalash fayliga misol:

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android=http://schemas.android.com/apk/res/android
    android:interpolator="@android:anim/accelerate_interpolator">
    <rotate
        android:fromDegrees="0"
        android:toDegrees="270"
        android:pivotX="50% "
        android:pivotY="50% "
        android:startOffset="500"
        android:duration="1000" />
    <scale
        android:fromXScale="1.0"
        android:toXScale="0.0"
        android:fromYScale="1.0"
        android:toYScale="0.0"
        android:pivotX="50% "
        android:pivotY="50% "
        android:startOffset="500"
        android:duration="500" />
    <alpha
        android:fromAlpha="1.0"
        android:toAlpha="0.0"
        android:startOffset="500"
```

```
android:duration="500" />
</set>
```

Qadamlab animatsiyalashni tavsiflaydigan resurs `res/drawable` katalogida saqlanadi. Navbatdagi misolda har biri 200 millisekund davomida aks etadigan (oxiridan tashqarida) poezdning oltita tasvirlarini ketma-ket aks ettirishga asoslangan animatsiyalash tavsiflangan. Tushunarliki, bunday animatsiyalashning ishlatilishi uchun bu katalogning o'zida *train1.png* .. *train6.png*. nomlarli (PNG formatdagi variant kabi) tasvirlarli (Drawable) resurslar kerak.

```
<animation-list
xmlns:android=http://schemas.android.com/apk/res/android
android:oneshot="false">
<item android:drawable="@drawable/train1" android:duration="200" />
<item android:drawable="@drawable/train2" android:duration="200" />
<item android:drawable="@drawable/train3" android:duration="200" />
<item android:drawable="@drawable/train4" android:duration="200" />
<item android:drawable="@drawable/train5" android:duration="200" />
<item android:drawable="@drawable/train6" android:duration="1500" />
</animation-list>
```

Menyu

Menyu resurslari ham Aktivliknin bosh menyusini uchun, ham foydalanuvchi interfeysi qandaydir elementini uzoq vaqt bosishda paydo bo'ladigan kontekst menyu (tushunarliki, agar bu qo'llansa va sizning ilovangiz uchun ma'noga ega) ishlatilishi mumkin. XML formatda tavsiflangan menyu `MenuInflater` tizim servisi *inflate* usuli yordamida ilovaga yuklanadi. Odatda bu Aktivlikda oldindan

aniqlangan *onCreateOptionsMenu* (bosh menyu uchun) yoki *onCreateContextMenu* (kontekst menyu uchun) usuli ichida bo‘lib o‘tadi. Menyuning har bir nusxasi res/menu katalogidagi alohida XML-faylda tavsiflanadi. Odatdagidek, fayllar nomlari (kengaytirishlarsiz) resurslar nomlari bo‘lib qoladi. Quyida uchta Refresh, Settings va Quit punktlarga ega bo‘lgan menyuli oddiy resursga misol keltirilgan.

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
  <item android:id="@+id/menu_refresh"
    android:title="Refresh" />
  <item android:id="@+id/menu_settings"
    android:title="Settings" />
  <item android:id="@+id/menu_quit"
    android:title="Quit" />
</menu>
```

Bu menyu punktlaridan har biri keyinchalik menyuni ishlab chiquvchiga foydalanuvchi punktlardan qaysi biri tanlanishini aniqlashga imkon beradigan noyob identifikatorga (*menu_refresh*, *menu_settings* i *menu_quit*) ega bo‘ladi.

Ilova kodidagi tashqi resurslardan foydalanish

Kodda resurslarga ulanish avtomatik generatsiyalanadigan R sinf, aniqrog‘i, uning nimsinflari yordamida aniqlanadi. Masalan, R sinf bizning loyihada quyidagi ko‘rinishga ega bo‘ladi:

```
package com.example.helloandroidworld;

public final class R {

  public static final class attr {
```

```

    }
    public static final class color {
        public static final int screen_bkg_color=0x7f040001;
        public static final int text_color=0x7f040002;
        public static final int view_bkg_color=0x7f040000;
    }
    public static final class drawable {
        public static final int ic_launcher=0x7f020000;
    }
    public static final class layout {
        public static final int main=0x7f030000;
    }
    public static final class string {
        public static final int app_name=0x7f050001;
        public static final int hello=0x7f050000;
    }
}

```

Resurslarga mos nomlarli sinflar a'zolari resurslarning o'z nusxalari bilan emas, balki resurslar jadvalidagi identifikatorlar hisoblanadi.

Ayrim usullar va konstruktorlar parametrlar sifatida resurslar identifikatorini qabul qilishi mumkin, bu holda ularni to'g'ridan-to'g'ri ishlatish mumkin:

```

setContentView(R.layout.main);
Toast.makeText(this,                                R.string.awesome_error,
Toast.LENGTH_LONG).show();

```

Agar resursning nusxasi zarur bo'lsa, *Resources* sinfi yordamida amalga oshiriladigan resurslar jadvaliga ulanish talab qilinadi. Bu sinf barcha resurslar turlari uchun getterlarga ega bo'ladi, bunda parametrlar sifatida R sinfdan resurslar identifikatorlari ishlatiladi:

```
// Resurslar jadvaliga ulanishni olamiz
Resources r = getResources();
// va resurslarning zarur nusxalarini olamiz
CharSequence greetingMsg = r.getText(R.string.greeting_message);
Drawable icon = r.getDrawable(R.drawable.app_icon);
int opaqueBlue = r.getColor(R.color.opaque_blue);
float borderWidth = r.getDimension(R.dimen.standard_border);
String[] stringArray = r.getStringArray(R.array.string_array);
int[] intArray = r.getIntArray(R.array.integer_array);
```

4. Hisobotning tarkibi

1. Ishlab chiqilgan loyihaning Xml fayli printskrinini keltirish.
2. Ishlab chiqilgan loyiha kodining listingini keltirish.

Nazorat savollari

1. O'lchamlarni tavsiflash uchun qaysi teg ishlatiladi?
2. O'lchamlarni tavsiflash uchun o'lchamlilik turlari qanday?
3. Uslubni yaratish uchun qaysi teg ishlatiladi?
4. *Drawable* resurslariga nimalar kiradi?
5. Belgilashli resurslar qaysi katalogda saqlanadi?

6. Android qaysi animatsiyalash turlari mavjud?
7. Qadamlab animatsiyalashni tavsiflaydigan resurs qaerda saqlanadi?
8. Qaysi usul yordamida ilovada XML formatda tavsiflangan Menyu yuklanadi?