№4-amaliyot ishi

Foydalanuvchi interfeysida boshqarish elementlaridan foydalanish

1. Ishdan maqsad

Amaliyot ishidan maqsad foydalanuvchi interfeysida turli boshqarish elementlari – tasvirli tugmalar, radiotugmalar, chekbokslar va boshqalardan foydalanishni o‘rganish

2. Topshiriq

1. Quyidagi tugmalarni yaratish:
   - Grafik;
   - Radiotugmalar;
   - Matnni kiritish uchun.
2. CheckBox elementni yaratish.
3. ToggleButton elementni yaratish

3. Amaliyot ishini bajarish bo‘yicha ko‘rsatmalar

1. YAngi ControlsSample loyihasini yarating.

2. res/layout/main.xml faylini faqat LinearLayout ildiz element qoladigan tarzda tahrir qiling. Unga keyinchalik zarur qo‘shimcha (dochernыy) elementlari qo‘shiladi:

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

```
<LinearLayout
xmlns:android=http://schemas.android.com/apk/res/android
android:layout_width="fill_parent"
android:layout_height="fill_parent"
android:orientation="vertical" >
</LinearLayout>
```

Grafik tugmalarning ishlatilishi

Tugmada matn o‘rnigatasvirning ishlatilishida tugmaning uchta holatlari – oddiy, tanlangan (“fokusda”) va bosilgan holatlari uchun uchta tasvirlar kerak bo‘ladi. Mos holatlari bu barcha uchta tasvirlar bunday tugmani yaratish uchun bitta XML faylda tavsiflanadi.

1. Tugmaning kerakli tasvirini res/drawable-mdpi katalogga ko‘chiring, Eclipseda katalog tarkibi ro‘yxatini yangilash uchun F5 tugmasini ishlatish mumkin.

2. Bu katalogning o‘zida tugmaning qaysi holatlarida qanday tasvirlarning ishlatish kerakligini tavsiflaydigan smile_button.xml faylni yarating:

```
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">
<item android:drawable="@drawable/smile_pressed"
android:state_pressed="true"/>
<item android:drawable="@drawable/smile_focused"
android:state_focused="true"/>
<item android:drawable="@drawable/smile_normal" />
</selector>
```

3. res/layout/main.xml belgilash faylidagi LinearLayout ichiga Button elementni qo‘shing:

```
<Button
android:id="@+id/button"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:background="@drawable/smile_button"
android:onClick="onButtonClicked"
android:padding="10dp" />
```

4. Aktivlikdan qaysi usul bu tugmaga bosilishga ishlov beruvchi sifatida ishlatilishini ko‘rsatadigan android:onClick="onButtonClicked" atributga e’tibor bering. Bu usulni Aktivlikka qo‘shing:

```
public void onButtonClicked(View v) {
Toast.makeText(this, "Tugma bosilgan", Toast.LENGTH_SHORT).show();
}
```

5. Ilovani ishga tushiring va tugmaning tasvirini turli holatlarga qanday o‘zgarishini, shuningdek tugmani bosilishiga ishlov bergich qanday ishlashini ko‘ring.

CheckBox vidjetining ishlatilishi

1. res/layout/main.xml belgilash faylidagi LinearLayout ichiga CheckBox elementni qo‘shing:

```
<CheckBox
android:id="@+id/checkbox"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:onClick="onCheckboxClicked"
android:text="Vыberi menya" />
```

2. android:onClick="onCheckboxClicked" atributi Aktivlikdan qaysi usul vidjetga bosishga ishlov bergich sifatida ishlatilishini aniqlaydi.
Bu usulni Aktivlikka qo‘shing:

```
public void onCheckboxClicked(View v) {
if (((CheckBox) v).isChecked()) {
Toast.makeText(this, "Otmecheno", Toast.LENGTH_SHORT).show();
} else {
Toast.makeText(this, "Belgilanmagan",
Toast.LENGTH_SHORT).show();
}
}
```

3. Ilovani ishga tushiring va turli holatlarda chekboksning o‘zini tutishini ko‘ring.

ToggleButton vidjetining ishlatilishi

Bu vidjet ikkita bir-birlarini inkor qiladigan holatlar, masalan, *YOqish/Uzish* oraida qayta ulanish talab qilinganida radiotugmalar va chekbokslarga muqobil sifatda yaxshi to‘g‘ri keladi.

1. res/layout/main.xml belgilash faylidagi LinearLayout ichiga ToggleButton elementni qo‘shing:

```
<ToggleButton android:id="@+id/togglebutton"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:textOn="Qo‘ng‘iroq yoqilgan"
android:textOff=" Qo‘ng‘iroq o‘chirilgan"
android:onClick="onToggleClicked"/>
```

2. android:onClick="onToggleClicked" atributi Aktivlikdan qaysi usul vidjetga bosishga ishlov bergich sifatida ishlatilishini aniqlaydi.

Bu usulni Aktivlikka qo‘shing:

```
public void onToggleClicked(View v) {
if (((ToggleButton) v).isChecked()) {
Toast.makeText(this, "Vklyucheno", Toast.LENGTH_SHORT).show();
} else {
Toast.makeText(this, "Vыklyucheno",
Toast.LENGTH_SHORT).show();
}
}
```

RadioButton vidjetining ishlatilishi

Radiotugmalar turli birlarini o‘zaro inkor qiladiganvariantlar orasida tanlash uchun ishlatiladi. Radiotugmalar guruhini yaratish uchun *RadioGroup* elementi ishlatiladi, uning ichida *RadioButton* elementlari joylashadi.

1. res/layout/main.xml faylidagi LinearLayout ichiga quyidagi belgilash elementlarini qo‘shing:

```
<RadioGroup
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:orientation="vertical" >
<RadioButton
android:id="@+id/radio_dog"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:onClick="onRadioButtonClicked"
android:text="Kuchukcha" />
<RadioButton
android:id="@+id/radio_cat"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:onClick="onRadioButtonClicked"
android:text="Mushukcha" />
<RadioButton
android:id="@+id/radio_rabbit"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:onClick="onRadioButtonClicked"
```

```
android:text="Quyoncha" />
</RadioGroup>
```

2. onRadioButtonClicked usulni Aktivlikka qo‘shing:

```
public void onRadioButtonClicked(View v) {
RadioButton rb = (RadioButton) v;
Toast.makeText(this, "Hayvon tanlangan: " + rb.getText(),
Toast.LENGTH_SHORT).show();
}
```

3. Ilovaning ishlashini tekshiring.

EditText vidjetining ishlatilishi

EditText vidjet foydalanuvchi tomonidan matnni kiritilishi uchun ishlatiladi. Bu vidjet uchun o‘rnatilgan tugmani bosilishiga ishlov bergich *Toast* yordamida kiritilgan matnni ko‘rsatadi.

1. res/layout/main.xml belgilash faylidagi LinearLayout ichiga EditText elementni qo‘shing:

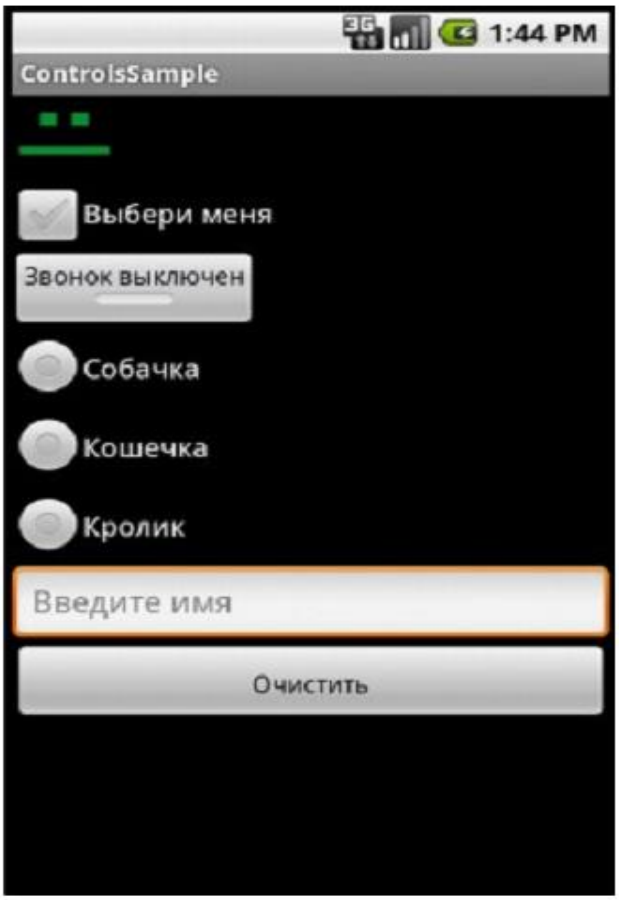
```
<EditText
android:id="@+id/user_name"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:hint="Nomni kiriting"/>
```

2. Foydalanuvchi kiritgan matnga ishlov berish uchun quyidagi kodni onCreate usulining oxiriga qo‘shing. E’tibor bering, bu ishlov bergich bz ishlatgan oldingilaridan farqli ravishda true yoki false qiymatlarni *qaytaradi*. Bu qiymatlarning semantikasi odatdagidek: *true* hodisaga (*event*) ishlov berilganligini va boshqa hech qanday amallar talab qilinmasligini bildiradi, *false* hodisaga bu ishlov bergich orqali ishlov berilmaganligini va znjirdagi boshqa ishlov bergichga uzatilishini bildiradi. Bizning holda aks ta’sir faqat Enter (*KEYCODE_ENTER*) tugmasini bosilishiga bo‘lib o‘tadi:

```
final EditText userName = (EditText) findViewById(R.id.user_name);
userName.setOnKeyListener(new View.OnKeyListener() {
@Override
public boolean onKey(View v, int keyCode, KeyEvent event) {
if ((event.getAction() == KeyEvent.ACTION_DOWN)
&& (keyCode == KeyEvent.KEYCODE_ENTER)) {
Toast.makeText(getApplicationContext(),
userName.getText(),
Toast.LENGTH_SHORT).show();
return true;
}
return false;
}
});
```

3. Ilovani ishga tushiring va uning ishlashini tekshiring.

4. Belgilashga “Tozalash” tugmasini qo‘shing va matnlar maydonini tozalaydigan ishlov bergichni yozing (*EditText* vidjetning *setText* usulini ishlating):



4.1-rasm. Belgilashga “Tozalash” tugmasini qo‘shilishi

5. Ilovaning ishlashini tekshiring.

4. Hisobning tarkibi

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

Nazorat savollari

1. Grafik tugmaning ishlash tartibini tushuntiring?
2. LinearLayout nima uchun ishlatiladi?
3. CheckBox elementi qanday funkiyani bajaradi?
4. android:onClick="onCheckboxClicked" atributi nimani aniqlaydi?
5. Qaysi hollarda ToggleButton vidjeti ishlatiladi?
6. RadioButton vidjetining funksional vazifalarini ko‘rsating?
7. EditText vidjetining vazifasini ayting?