

№5-amaliyot ishi

Menyuni yaratish va undan foydalanish

1. Ishdan maqsad

- Android-ilovalarni ishlab chiqish instrumentlari bilan tanishish;
- asosiy va matnbop menyuni yaratishni o‘rganish.

2. Topshiriq

1. MetroPicker paketni modifikatsiyalash;
2. Aktivlikda asosiy menyuni yaratish;
3. *TextView* Taqdim etish uchun matnbop menyuni dinamik yaratish;
4. Asosiy Aktivlik uchun asosiy menyuni yaratish.

3. Amaliyot ishini bajarish bo‘yicha ko‘rsatmalar

MetroPicker loyihasini quyidagi tarzda modifikatsiyalang:

1. Aktivlikka metro stansiyalarining ro‘yxatini aks ettiradigan asosiy menyuni qo‘shing
2. Menyuda “Qaytish” punkti bo‘lishi kerak. Menyuni kodda, satr resurslaridan foydalanmasdan dinamik yarating.
3. Asosiy Aktivlik tanlangan metr stansiyasini aks ettiradigan *TextView* Taqdim etish uchun matnbop menyuni dinamik yarating. Menu punktini tanlash tanlangan stansiyanı bekor qilishi kerak.

4. Asosiy Aktivlik uchun ikkita «bekor qilish» va «chiqish» punktlaridan asosiy menyuni yarating. Bu punktlarni tanlashda kerakli funksiyalarni ishlatib.

Androiddagi menyu

Menyu istalgan ilovaning muhim qismi hisoblanadi. Android tizimi standartlashtirilgan amaliy menyularni yaratish va ulardan turli xil funktsionallikdagi ilovalarda foydalanish uchun etarlicha oddiy interfeysni taqdim etadi.

Ilovalarda menyuning ishlatilishi foydalanuvchi interfeysining nisbatan kam ishlatiladigan elementlari bilan band bo'ladigan ekranning qimmatli maydonini saqlashga imkon beradi.

Har bir Aktivlik faqat uning uchun o'ziga xos funksiyalarni bajaradigan menyuga ega bo'lishi mumkin. Shuningdek ekrandagi har bir Taqdim etish uchun individual bo'lgan *matnbop* menyuni ham ishlatish mumkin.

Menyudan foydalanish asoslari

Androidda, birinchi navbatda, uncha katta bo'lmagan ekranlar uchun optimallashtirilgan uchta darajali menyu tizimini qo'llash bajarilgan:

– **Asosiy menyu** qurilmaning «menu» tugmasi bosilganida ekranning pastida paydo bo'ladi. U cheklangan punktlar soni (yashirish bo'yicha oltitadan oshmaydigan) uchun matn yoki belgilarni aks ettirishi mumkin. Bu menyu uchun relief elementlarili kulrang tusli ko'rinishdagi ranglar gammasili belgilardan foydalanish tavsiya etiladi. Bu menyu radiotugmalar va chekbokslarga ega bo'la olmaydi. Agar bunday menyuning punktlari soni maksimal ruxsat etiladigan qiymatdan oshsa menyuda «yana» («more») yozuvili punkt avtomatik paydo bo'ladi. unga bosilganda kengaytirilgan menyu aks etadi;

- **Kengaytirilgan menyu.** Kengaytirilgan menyu elementlari asosiy menyuga kirmaydigan punktlar hisoblanadigan o'tkaziladigan ro'yxatni aks ettiradi. Bu ro'yxatda belgilarni aks ettirish mumkin emas, lekin radiotugmalar va chekbokslarni aks ettirish imkoniyati mavjud. Madomiki, asosiy menyu o'rniga kengaytirilgan menyuni aks ettirish usuli mavjud emas ekan, ilova yoki tizimning qandaydir komponentlarini o'zgarishi haqida foydalanuvchini menyu belgilari yoki punktlar matnini o'zgarishi yordamida ogohlantirish tavsiya etiladi;

- **Qo'shimcha menyu** (uchinchi daraja menyusi) asosiy yoki kengaytirilgan menyudan chaqirilishi va qalquchi oynada aks ettirilishi mumkin. Qo'shilganlik qo'llanmaydi va qo'shimcha menyudan yana bir menyuni chaqirishga urinish inkor etishni bekor qilinishiga olib keladi.

Menyuni yaratish

Menyuga murojaat qilinganida Aktivlik *onCreateOptionsMenu* usuli chaqiriladi va menyuni ekranda paydo bo'lishi uchun uni oldindan aniqlash talab qilinadi. Bu usul parametr sifatida *Menu* sinfidagi parametrni oladi, u keyinchalik menyu punktlari bilan amallarni bajarish uchun ishlatiladi.

Menyuga yangi elementlarni qo'shish uchun *Menu* ob'ektining quyidagi parametrlarli *add* usuli ishlatiladi:

- **Guruh:** guruhli ishlov berish uchun menyu punktlarini birlashtirish uchun;
- **Identifikator:** menyu punkti noyob identifikatori. Bu t identifikator menyu punktiga ishlov beruvchiga - *onOptionsItemSelected* usuliga uzatiladi.
- **Tartib:** menyu punktlari chiqariladigan tartibni ko'rsatadigan qiymat.
- **Matn:** menyuning bu punktidagi yozuv.

Menyu muvaffaqiyatli yaratilganidan keyin *onCreateOptionsMenu* usuli *true* qiymatni qaytarishi kerak. Misol satr resurslaridan foydalaniladigan uchta punktlardan menyuni yaratilishini ko'rsatadi:

@Override

```
public boolean onCreateOptionsMenu(Menu menu) {  
    super.onCreateOptionsMenu(menu);  
    menu.add(0, Menu.FIRST, Menu.NONE, R.string.menu_item1);  
    menu.add(0, Menu.FIRST+1, Menu.NONE, R.string.menu_item2);  
    menu.add(0, Menu.FIRST+2, Menu.NONE, R.string.menu_item3);  
    return true;  
}
```

Identifikator bo'yicha menyu punktlarini qidirish uchun *Menu* ob'ektining *findItem* usulidan foydalanish mumkin.

Menyu punktlari parametrlari

Menyu punktlari eng foydali parametrlari quyidagilar hisoblanadi:

- ***Qisqacha sarlavhalar***: agar punkt asosiy menyuda aks ettirilishi mumkin bo'lsa ishlatiladi. *MenuItem* sinfi ob'ektining *setTitleCondensed* usuli orqali o'rnatiladi: `menuItem.setTitleCondensed ("Sarlavha");`
- ***Belgilar***: ***Drawable identifikatori***, u kerakli tasvirga ega bo'ladi: `menuItem.setIcon(R.drawable.menu_item_icon);`
- ***menyu punktni tanlashga ishlov bergich***: o'rnatish mumkin, lekin unumdorlikni oshirish nuqtai nazaridan tavsiya etilmaydi, yaxshisi butun menyuga ishlov bergichni (*onOptionsItemSelected*) ishlatish kerak. SHunga qaramay, misol keltiramiz:

```
menuItem.setOnMenuItemClickListener(new OnMenuItemClickListener() {  
    public boolean onMenuItemClick(MenuItem _menuItem) {
```

```
// punktni tanlashga ishlov berish  
return true;  
}  
});
```

- **Maqsad:** bu maqsad, agar menyu punktiga bosish *onMenuItemClickListener* va *onOptionsItemSelected* ishlov bergich orqali ishlov berilmagan bo'lsa, *startActivity* usuliga avtomatik uzatiladi:

```
menuItem.setIntent(new Intent(this , MyOtherActivity.class));
```

Menyu punktlarini dinamik o'zgartirish

Menyuni ekranga to'g'ridan-to'g'ri chiqarishdan oldin joriy Aktivlikning *onPrepareOptionsMenu* usuli chaqiriladi va uni oldindan aniqlash bilan menyu punktlarining holatlarini dinamik o'zgartirish - ruxsat etish/ta'qiqlash, ko'rinmaydigan qilish, matnni o'zgartirish mumkin va h.k.. Modifikatsiyalanishi kerak bo'lgan menyu punktini qidirish uchun parametr sifatida uzatiladigan *Menu* ob'ektining *findItem* usulini ishlatish mumkin.

```
@Override  
public boolean onPrepareOptionsMenu(Menu menu) {  
    super.onPrepareOptionsMenu(menu);  
    MenuItem menuItem = menu.findItem(MENU_ITEM);  
    //menyu punktini modifikatsiyalash....  
    return true;  
}
```

```
}
```

Menyu punktini tanlashga ishlov berish

Android bitta *onOptionsItemSelected* ishlov bergich bilan menyuning barcha punktlariga (ularni tanlanishiga) ishlov berishga imkon beradi. Tanlangan menyu punkti bu ishlov beruvchiga *MenuItem* sinfi ob'ekti sifatida uzatiladi.

Menyu punktini tanlashga kerakli reaksiyani ishlatilishi uchun nima aynan talanganini aniqlash talab qilinadi. Buning uchun ob'ekt parametri sifatida uzatilgan *getItemId* usuli ishlatiladi, olingan natija esa *onOptionsItemSelected* usulida menyuga punktlarni qo'shishda ishlatiladigan identifikatorlar bilan solishtiriladi.

```
@Override
```

```
public boolean onOptionsItemSelected(MenuItem item) {  
    switch (item.getItemId()) {  
        // Har bir ma'lum punktni tekshirish  
        case (MENU_ITEM):  
            // nimadir qilish...  
            return true;  
    }  
    // Agar punktga ishlov berilmagan bo'lsa, uni ishlov berishga uzatish  
    return super.onOptionsItemSelected(item);  
}
```

Qiz va matnbop menyular

Qiz va matnbop menyular ekranda paydo bo'lganida qalquvchi oynalar ko'rinishida bir xil ko'rinadi, lekin bunda ular turlicha yaratiladi.

Qiz menyuni yaratish

Qiz menyuni yaratish uchun *Menu* sinfi *addSubMenu* usuli ishlatiladi:

```
SubMenu sub = menu.addSubMenu(0, 0, Menu.NONE "qiz menyu");  
sub.setHeaderIcon(R.drawable.icon);  
sub.setIcon(R.drawable.icon);  
MenuItem submenuItem = sub.add(0, 0, Menu.NONE, "qiz menyu punkti");
```

YUqorida aytilganidek, qo'yilgan qiz menyularni Android qo'llamaydi.

Matnbop menyuni yaratish

Androidda matnbop menyuni yaratishning eng ko'p ishlatiladigan usuli Aktivlikning *onCreateContextMenu* usulini oldindan aniqlash hisoblanadi:

@Override

```
public void onCreateContextMenu(ContextMenu menu, View v,  
ContextMenu.ContextMenuInfo menuInfo) {  
    super.onCreateContextMenu (menu, v, menuInfo);  
    menu.setHeaderTitle("Konekst menyu");  
    menu.add(0, Menu.FIRST, Menu.NONE, "1-punkt");  
    menu.add(0, Menu.FIRST+1, Menu.NONE, "2-punkt");  
    menu.add(0, Menu.FIRST+2, Menu.NONE, "3-punkt");  
}
```

Kerakli Taqdim etishlar uchun matnbop menyuga ishlov bergichni qayd etish *registerForContextMenu* Aktivlik usuli yordamida amalga oshiriladi:

@Override

```

public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);
    tv = (TextView) findViewById(R.id.text_view);
    registerForContextMenu(tv);
}

```

XML yordamida menyuni tavsiflash

Ko‘pincha menyuni resurslar ko‘rinishida ierarxik tavsiflash ham qulay bo‘ladi. bunday yondashishning afzalliklari haqida yuqorida aytilgan. Menyu loyihaaning res/menu katalogida an’anaviy tavsiflanadi va saqlanadi. Menyuning butun ierarxiyasi (agar ierarxik menyu mavjud bo‘lsa) alohida fayllarda joylashishi, faylning nomi esa resurs identifikatori sifatida ishlatilishi kerak. Faylning ildiz elementi `<menu>` tegi bo‘lishi kerak, menyu punktlari esa `<item>` tegi orqali tavsiflanishi, menyu punktlari Xossalari quyidagi mos atributlarorqali tavsiflanishi kerak:

```

<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android" >
  <item
    android:id="@+id/item01"
    android:icon="@drawable/menu_item"
    android:title="1-punkt">
  </item>
  <item
    android:id="@+id/item02"
    android:checkable="true"

```



```
    android:title="2-punkt">
</item>
<item
    android:id="@+id/item03"
    android:title="3-punkt">
</item>
<item
    android:id="@+id/item04"
    android:title="1-qiz menyu">
<menu>
<item
    android:id="@+id/sub1item01"
    android:title="1-qiz menyu punkti">
</item>
</menu>
</item>
<item
    android:id="@+id/item05"
    android:title="2-qiz menyu ">
<menu>
<item
    android:id="@+id/sub2item01"
    android:title="2-qiz menyu punkti">
</item>
</menu>
</item>
</menu>
```

onCreateOptionsMenu va *onCreateContextMenu* hodisalarida resurslardan Menu ob'ektlarni yaratish uchun *MenuInflater* turdagi ob'ekt *inflate* usuli hisoblanadi::

```
@Override  
public boolean onCreateOptionsMenu(Menu menu) {  
    super.onCreateOptionsMenu(menu);  
    MenuInflater inflater = getMenuInflater();  
    inflater.inflate(R.menu.menu1, menu);  
    return true;  
}
```

yoki:

```
@Override  
public void onCreateContextMenu(ContextMenu menu, View  
v, ContextMenuInfo menuInfo) {  
    super.onCreateContextMenu(menu, v, menuInfo);  
    MenuInflater inflater = getMenuInflater();  
    inflater.inflate(R.menu.menu2, menu);  
    menu.setHeaderTitle("Matnbop menyu");  
}
```

4. Hisobning tarkibi

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

Nazorat savollari

1. *Menu* tshunchasini bering?
2. Menyuning uchta bosqichli tizimi haqida soʻzlang?
3. Asosiy menu nima hisoblanadi?
4. Kengaytirilgan menu nimalarni oʻz ichiga oladi?
5. Qoʻshimcha menu deb nimaga aytiladi?
6. *Menu* obʼekti usuli *add* nima uchun ishlatiladi va u qaysi parametrlarni oʻz ichiga oladi?
7. Menu punktlarning foydali parametrlarini sanab oʻting?
8. Qaysi usul yordaida menu punktlarini dinamik oʻzgartirish mumkin?
9. Qaysi ishlov bergich menu punktlariga ishlov berishga imkon beradi?
10. Qoʻshimcha menu qanday yaratiladi?
11. Matnbop menu qanday yaratiladi?