

Vaje 2019/2020 Platformno odvisen razvoj aplikacij

16. december 2019

1 Vaje 1: Java basics

Cilj vaj je spoznati osnove Jave, ki so potrebne za razvoj aplikacij Android.

1.1 Primer javnaskega projekta

Na GitHub spletišču poiščite dva Javanska projekta in v njih dva razreda. Poizkusite razumeti kodo. [Projekta lahko najdete tukaj](#).

1.2 Java razredi

Napišite poljuben razred, ki ima vsaj tri lastnosti različnih tipov (int, double, string ...). Vaš razred mora implementirati razred Comparable. V metodi compareTo primerjajte objekta po poljubni lastnosti. Povežite tudi metodo toString(). V main metodi napolnite seznam z vsaj petimi objekti vašega razreda, seznam nato sortirajte in izpišite.

1.3 Java interface (vmesnik)

Napišite vmesnik Sizable z abstraktno metodo size(), ki vrača dolžino (int). Na prejšnjem primeru demonstriraj uporabo vmesnika.

1.4 Javanske konvencije

Na spletu poišči Javanske konvencije pisanja izvirne kode.

Odgovorite najmanj na:

- Kako se pišejo razredi?
- Kako se pišejo lastnosti?
- Kako se pišejo konstante?
- Kako se pišejo metode?

1.5 Izjeme v javi

Poiščite vsaj 3 Javanske izjeme ter demonstrirajte njihovo uporabo. Napišite lastno izjemo (deduje iz Exception) in pripravite primer, kjer jo prožite in lovite.

2 Vaje 2: izdelava grafičnega vmesnika GUI

Izdelava grafičnega vmesnika sestoji iz razvrščevalnika (layout) in grafičnih komponent.

2.1 LinearLayout

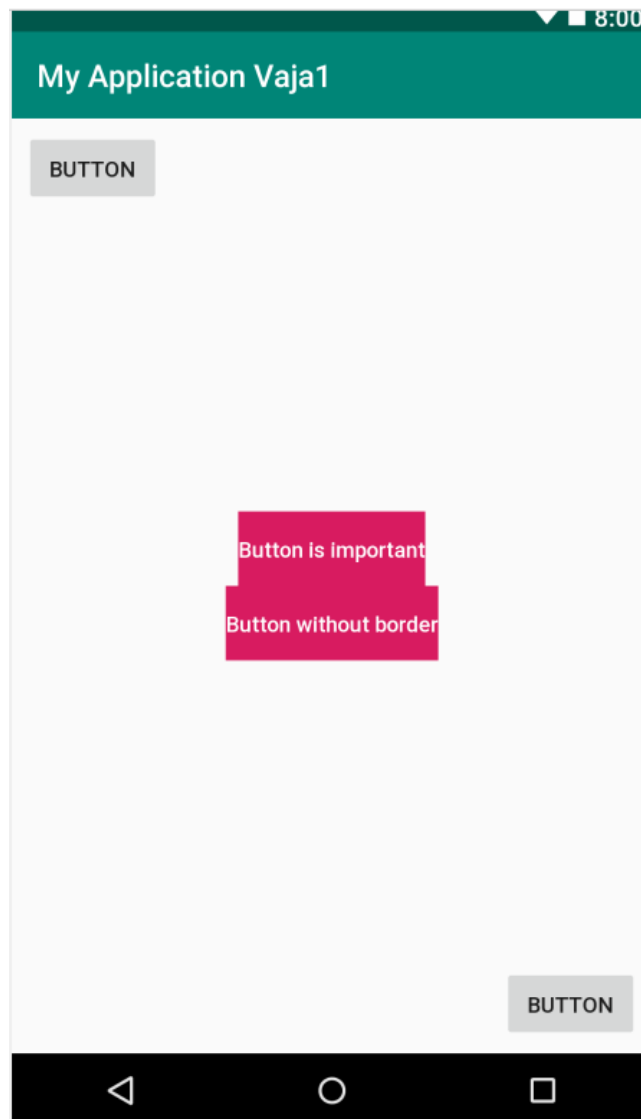
Naredite grafičen vmesnik z uporabo LinearLayout-ov, ki bo imel razporeditev čim bolj identično sliki 1.



Slika 1: Primer vmesnika izdelanega z uporabo LinearLayout-ov.

2.2 ConstraintLayout

Naredite grafičen vmesnik z uporabo ConstraintLayout-a, ki bo imel razporeditev čim bolj identično sliki 2. Pri izdelavi vmesnika bodite pozorni na to, da je korenski element ConstraintLayout in da ne gnezdite [elementov](#).



Slika 2: Primer vmesnika izdelanega z uporabo ConstraintLayout-a.

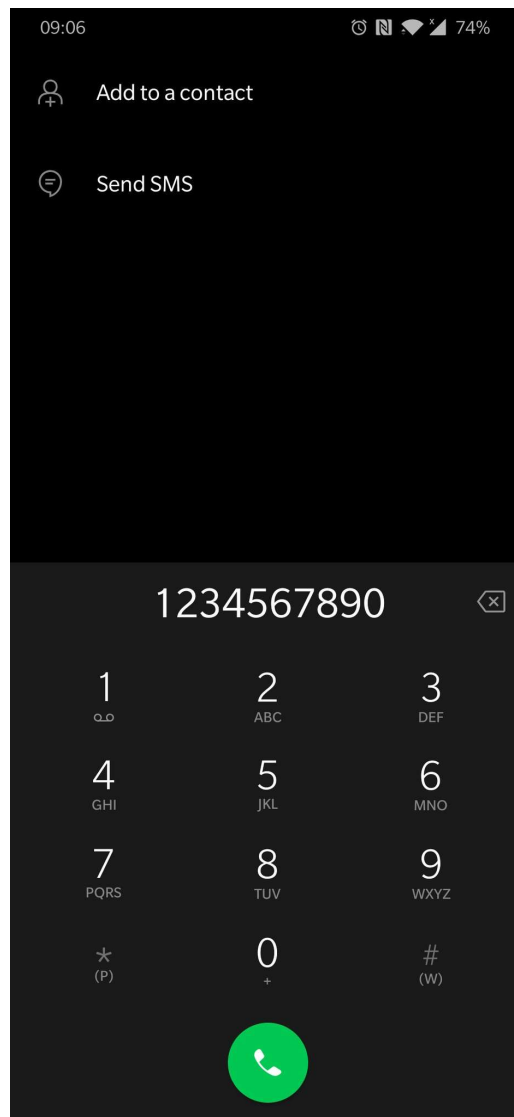
2.3 Prijavni zaslon (login)

Na spletu poiščite sliko (screenshot) prijavnega zaslona. Prijavni zaslon naj vsebuje logo (slika), polji za vnos uporabniškega imena in gesla, ter gumb za prijavo. Nato naredite grafičen vmesnik prijavnega zaslona, ki bo čim bolj podoben izbrani sliki. Pri tej vaji izdelajte samo vmesnik! **Pri izdelavi vmesnika je obvezna uporaba ConstraintLayout-a!**

Za kaj bi uporabili referenčno črto (baseline)?

2.4 Zaslon za vnos telefonske številke (phone dial screen)

Izdelajte vmesnik za vnos telefonske številke. Pri zgledu vmesnika imate proste roke. Lahko se zgledujete po vmesniku iz vašega telefona, slike iz spleta, ali pa izdelate popolnoma unikaten vmesnik. Pri izdelavi vmesnika je obvezna uporaba ConstraintLayout-a! Primer vmesnika lahko vidite na sliki [3](#).



Slika 3: Primer vmesnika za vnos telefonske številke.

3 Dogodki

Nalogo iz poglavja 2.4 dopolnite:

- Napišite razred Call, ki vsebuje telefonsko (string) in čas klica.
- Napišite razred Calls, ki vsebuje seznam klicev (objekti Call). Primerek liste z imenom "Trenutni" ustvari ob zagonu aplikacije.
- Na klik gumba kliči se izpiše Toast, kjer piše "Kličem" in trenutna telefonska številka. Prav tako ustvarite objekt Call in ga dodajte v listo in jo izpišite celotno v log (namig: Log.i(...)).
- Implementirajte gumb za brisanje zadnje cifre. V primeru dolgega klika izbrišete vse.
- Napišite novo aktivnost "O avtorju" (AboutAuthorActivity), kjer so prikazani vaši podatki (slika, ime, priimek...). Aktivnost se odpre ob dolgem kliku na zvezdico.
- Ob zapiranju (vračanju) iz aktivnosti "O avtorju" se vrne *vaša* telefonska številka, ki se izpiše na zaslon (namig: startActivityForResult).

4 Viri UUID

Nalogo iz poglavja 2.4 dopolnite:

- UUID se ustvari ob prvem zagonu aplikacije. Ob naslednjih zagonih se prebere iz diska (SharedPreferences).
- Uredi ikono za zagon in podpri različne ločljivosti.
- Številčnico opremi s posebno pisavo (najdi jo na spletu).
- Aplikacija naj vsebuje vsaj eno vektorsko sliko.

5 Vodenje stvari

Tokrat boste napisali novo aplikacijo, ki bo omogočala vodenje stvari. Katere stvari boste vodili si izmislite sami. Npr. število prevoženih km, obiskov v fitnesu, pridelek na vrtu, število košev na tekmi, ... Poizkusite biti izvirni.

V okviru te naloge naredite:

- Javanski "Data" razredi naj bodo v svoji knjižnici (javanski modul libdata).
- Aktivnost vnosa ali popravljanja "stvari". Vsaka stvar vsebuje interno šifro UUID, naziv, datum vnosa, itd... Gumb Shrani, ...
- Aktivnost izbira, ki ima na vrhu spinner, kjer se izbere ena izmed "stvari". Po izbiri se spodaj izpišejo podatki o stvari.
- Implementirajte svoj razred Application, kjer je vodite stanje aplikacije (glej primer iz predavanj).
- Vse se naj s pomočjo serializacije JSON shrani (uporabite GSON).
- Uporabite nov git repozitorij (PORA A2 Vodenje stvari).

6 Vodenje stvari (nadaljevanje)

Nalogo dopolnite tako, da demonstrirate uporabo razreda RecyclerView. Implementirajte aktivnost seznam, ki prikaže vse podatke o stvareh v več vrsticah in je lepo formatiran. Pripravite metodo, ki vam bo generirala stvari (elemente seznama). To metodo nato uporabite, da napolnite seznam z vsaj 100 elementi.

- Na klik "stvari" si izmislite neko funkcionalnost (npr. zaloga se poveča za 1).
- Na dolgi klik implementirajte brisanje. Prej uporabnika povpraša s pomočjo dialoga DA/NE.
- Na aktivnost dodajte kakšno statistično informacijo (povprečje, min, max, mediana, ...).

7 Vodenje stvari (nadaljevanje 2. del)

Cilj naloge je uporaba knjižnice EventBus za komunikacijo med aktivnostmi in storitvami, ter implementacija lastne storitve. Za osnovo vzemite primer iz predavanj.

* Ne pozabite dodati pravice za GPS podporo v datoteko AndroidManifest.xml!

* Ob zagonu aplikacije preveriti ali je uporabnik odobril pravico za uporabo GPS (checkSelfPermission).

- Aplikacijo spremenite tako, da bo na vseh aktivnostih na vrhu odsek z besedilom, ki izpisuje trenutne GPS koordinate (latitude in longitude). V primeru, da ni GPS signala se z rdečo barvo izpiše tekst no GPS signal, v nasprotnem primeru se izpišejo koordinate z zeleno barvo.
- Vsaka stvar naj bo po novem opremljen še z lokacijo (latitude in longitude) in zastavico hasLocation. Če lokacija ni na voljo se nastavi zastavica hasLocation na false in vrednosti lat. in long. na 0.
- Stvar, ki jo beležite dodajte še lastnosti insertedTime (čas vnosa) in updatedTime (čas spremembe).
- Za komunikacijo med servisom in aplikacijo oz. aktivnostmi uporabljaj knjižnico EventBus.
- Neobvezen del. Če GPS med delovanjem aplikacije "pade", se ustrezno osveži status.
- Neobvezen del. V odseku se izpisuje še pot od točke, ko je bil prvič zaznan GPS in do trenutne točke (razdalje morate sproti seštevati).

8 Prestrezanje sistemskih dogodkov

Aplikacijo vodenje stvari spremenite tako, da lahko s pomočjo SMSa vnesete novo stvar. To naredite tako, da kadar ste v aplikaciji in dobite SMS in ta SMS vsebuje ključne besede (in podatke) samodejno vnesete nov podatek. [Primer poslušanja SMSov](#).

8.1 Dodatna

Podobno kot prej še implementirajte dva izbrana dogodka ([Seznam](#)), za katera si izmislite funkcionalnosti.

9 View po meri

Vzami primer iz predavanj (MyClock) in ga spremeni tako, da bo imel okoli komponente izrisan okvir, ki mu lahko določiš barvo (v XML layout).

9.1 Dodatna

Naredi svojo komponento, ki ti izboljša program stvari (ta del naloge ni časovno omejen).

Primer

- V listi, se ob vsaki stvari izriše puščica, ki kaže smer stvari (lokacija kjer je stvar vnešena) glede na trenutni položaj.
- Za namen prikaza statistike si izmisliš nek graf.

10 Knjižnice/orodja

Android in java sta ena izmed najbolj odprtih skupnosti. Na spletu najdemo ogromno koristnih knjižnic in orodij. Poišči zanimivo Android knjižnico ali orodje, ter jo/ga uporabi v enem izmed svojih projektov (največ trije lahko imajo enako knjižnico, knjižnice iz predavanj so izvzete oz. niso dovoljene). V ta namen se vpišite v google razpredelnico. Pripravite predstavitev od 5 do 10min (7-15 strani). Vsebuje naj vsaj spodnje strani:

- **Kaj?** Opiši kaj dela. Kdaj je nastala, popularnost, aktivnost projekta...
- **Scenariji uporabe.** Naštej nekaj scenarijev, ko bi uporabil knjižnico.
- **Zakaj?** Zakaj knjižnico, kako bi to naredil brez knjižnice.
- **Kako?** Demonstrirati uporabo. Kodo,...
- **Možne težave,** pri uporabi. Kaj vam ni delalo iz prve.
- **Osebno mnenje,** kritični pogled, zahtevnost uporabe, hitrost, prostorska zahtevnost...

Predstavitev boste predstavili takoj po praznikih.