

# KI205 – Android

## TEST 1

1. Koje verzije Android OS-a poznajete?

- 1.0
- 1.1
- 1.5 – Cupcake
- 1.6 – Donut
- 2.0 – Eclair
- 2.2 – Froyo
- 2.3 – Gingerbread
- 3.0 – Honeycomb
- 4.0 – Ice Cream Sandwich
- 4.1 – Jelly Bean
- 4.4 – KitKat
- 5.0 – Lollipop
- 6.0 – Marshmallow
- 7.0 – Nougat
- 8.0 – Oreo

2. Šta čini jezgro Android OS-a?

- Linux jezgro

3. Navesti nekoliko karakteristika Android OS-a?

- Unificirani pristup razvoju aplikacija,
- Pristupanje mobilnim mrežama putem svih poznatih mobilnih standarda (GSM, GPRS, EDGE, UMTS, LTE),
- Umrežavanje putem: Bluetooth, NFC, WiFi, WiMAX,
- Slanje tekstualnih i multimedijalnih poruka,
- Web čitač,
- Multitasking,
- Podrška za Touchscreen.

4. Koji slojevi čine Android OS? Objasniti u kratkim crtama.

- **Linux jezgro** – je osnova OS-a i u njemu su sadržani pokretački programi svih hardverskih komponenata integrisanih u mob. uređaj.
- **Biblioteke** – čuvaju prog. kodove svih modula neophodnih za funkcionisanje OS-a.
- **Android Runtime** – je istog nivoa kao i biblioteke i obezbeđuje arhive pomoću kojih Java programeri pišu Android aplikacije.
- **Radni okvir** – (Framework), obezbeđuje različite funkcionalnosti Android OS-a bez kojih kreirane aplikacije ne bi mogle da se izvršavaju.
- **Nivo aplikacije** – obuhvata sve aplik. koje su pre-instalirane na uređaju.

5. Šta je određeno AndroidManifest.xml datotekom a šta activity\_main.xml datotekom?

- **AndroidManifest.xml** – je datoteka u kojoj su definisane privilegije neophodne za aplikaciju, navigacija, filteri sadržaja, ... i sl.
- **activity\_main.xml** – je datoteka u kojoj je definisan korisnički interfejs aplikacije.

6. Objasniti arhitekturu Android aplikacije?

- **Arhitektura** Android aplikacije je realizovana pomoću foldera i datoteka koje su od značaja.

U najznačajnije **foldere** spadaju:

- **src** – sadrži paket sa izvornim datotekama sa ekstenzijom .java. U ovim datotekama je sadržan programski kod same aplikacije.
- **gen** – sadrži R.java datoteku, koju je kreirao Java prevodilac i ona upućuje na sve resurse u okviru projekta.
- **Android x.x** – je folder koji sadrži datoteku android.jar koja predstavlja Java arhivu sa svim klasama neophodnim za razvoj Android aplikacije.
- **assets** – sadrži sve neophodne alate koje će aplikacija koristiti, kao, npr: HTML, tekstualne datoteke, bazu podataka, ... i sl.
- **bin** – sadrži sve datoteke koje su nastale angažovanjem ADT-a u procesu prevođenja aplikacije. Ovde se nalazi i .apk fajl, koji predstavlja binarni kod aplikacije.
- **res** – je folder u kome su sadržani svi resursi koji se koriste u aplikaciji.

U najznačajnije **datoteke** spadaju:

- **AndroidManifest.xml** – je datot. u kojoj su def. aktivnosti aplikacije.
- **activity\_main.xml** – služi za def. korisničkog interfejsa.
- **MainActivity.java** – metodom setContentView() povezuje korisnički interfejs sa aktivnošću.
- **R.java** – je datoteka automatski ažurirana od strane Eclipse IDE-a

7. Kako se testira kreirana Android aplikacija?

- Kreirana Android aplikacija se testira puštanjem aplikacije u rad na emulatoru (AVD uređaju) ili na realnom mob. uređaju.

8. Objasniti postupak kreiranja AVD instance?

- **AVD** je instanca emulatora koja omogućava modeliranje realnog uređaja. Kreira se pomoću ADV Manager-a u radnom okruženju Android Studia.

9. Navedite neophodne tehnologije za razvoj Android aplikacija?

- U neophodne alate i tehnologije spadaju: JAVA JDK SE, Android SDK, Eclipse IDE (ili Android Studio), ADT (Android Development Tools).

10. Šta je Android SDK?

- **Android SDK** je softverski paket koji sadrži debugger, emulator, dokumentaciju, primere koda i uputstva. Koristi se za kreiranje Android aplikacija.

11. Šta se postiže konfigurisanjem Android SDK menadžera?

- Postiže se mogućnost upravljanja različitim verzijama Android SDK-a instaliranim na računaru.

12. Šta je Eclipse IDE i zašto se veoma široko koristi?

- **Eclipse IDE** je najzastupljenije razvojno okruženje u industriji softvera. Koristi se mnogo zbog svoje proširivosti zahvaljujući mnoštvu plug-in-ova.

13. Kako se naziva zvanično radno okruženje kompanije Google za razvoj mobilnih aplikacija?

- **Android Studio**

14. Navesti i objasniti opcije koje su ponuđene pokretanjem Android Studio IDE razvojnog okruženja?

- Ponuđene opcije su:
  - Izbor novog Android projekta,
  - Otvaranje postojećeg Android projekta,
  - Importovanje postojećih Android projekata
  - Importovanje projekata iz drugih IDE,
  - Provere projekata,
  - Podešavanja,
  - Dokumentacija.

15. Objasnit postupak kreiranja novog Android projekta u Android Studio IDE razvojnom okruženju?

- Klikom na **Start new Android Studio Project** započinje se razvoj novog Android projekta. Zatim se vrše podešavanja parametara novog projekta kao što su definisanje naziva aplikacije i paketa u kome će se aplikacija nalaziti, definisanje lokacije na hard-disku, izbor uređaja kojem je aplikacija namenjena, izbor minimalnog SDK za koji se aplikacija razvija. Kao poslednje bira se ponuđeni šablon za klasu aktivnosti i na kraju se klikne dugme **Finish**.

16. Objasnite proceduru kreiranja ADV instance u Android Studi IDE razvojnom okruženju?

- U Android Studio IDE okruženju, bira se opcija **ADV Manager**, kojom je omogućeno kreiranje novog Android emulatora. U prozoru **Android Virtual Device** je omogućeno pokretanje postojećeg emulatora, njegovo modifikovanje ili kreiranje novog. Izborom opcije **New Hardware Profile** definiše se novi AVD uređaj. NA kraju se definiše Android API za kreirani emulator, a u prozoru **Verify Configuration** se obavlja završetak definicije kreiranog AVD-a. Na kraju se klikne na dugme **Finish**.

17. Da li je moguće importovati Android projekat iz Eclipse IDE u Android Studio IDE? Objasniti.

- Moguće je. To se radi u sledećim koracima:
  - Prvo se vrši priprema projekta za eksportovanje iz Eclipse IDE. Bira se: **File** -> **Export**.
  - Zatim se kreira **Gradle Build Files**, kojima će biti omogućeno eksportovanje datoteka iz Eclipse IDE u Android Studio.
  - Zatim se u Android Studiju biraju: **File** -> **New** -> **Import Project**.
  - Zatim se bira konkretan projekat koji se importuje u Android Studio, navigacijom do njegove lokacije na disku.
  - Nakon izbora projekta i klika na **OK**, Android Studio započinje importovanje izabranih datoteka.

18. Koju VM koristi Gennymotion za pokretanje vlastite AVD instance?

- Oracle VM Virtual Box

19. Šta je neophodno učiniti za korišćenje Gennymotion AVD instance u Eclipse IDE i Android Studio razvojnim okruženjima?

- Potrebno je izvršiti njihovo povezivanje sa odgovarajućim razvojnim alatom, instalacijom odgovarajućih dodataka plug-in-ova.

20. U koju vrstu datoteka se dodaju aktivnosti? \*\*\*

- **AndroidManifest.xml**. Znači u .xml datoteci.

21. Šta promena orijentacije ekrana izaziva? \*\*\*

- Svaki put kada se orijentacija ekrana promeni poziva se metoda **onCreate()** za iniciranje aktivnosti. Desiće se prekid trenutne aktivnosti.

22. Android uređaju podržavaju sledeće orijentacije ekrana? \*\*\*

- **Portrait i Landscape.**

23. Kako se Android aplikacija bori sa sudaranjem filtera namere? \*\*\*

- Prikazuje (poziva) selektor aktivnosti

24. Kojom datotekom je definisan korisnički interfejs Android aplikacije? \*\*\*

- **activity\_main.xml**

## TEST 2

1. Šta je aktivnost (Activity) i kako se implementira?

- **Aktivnost** (Activity) predstavlja jedan ekran sa korisničkim interfejsom. Svaka aktivnost se implementira kao podklasa klase **Activity** ili **AppCompatActivity**.

2. U kojoj datoteci i na koji način se inicijalizuje aktivnost u Android aplikaciji?

- Svaka aktivnost aplikacije mora da bude deklarirana u **AndroidManifest.xml** datoteci. Inicira se pozivanjem **onCreate()** metode, jer se njome realizuje učitavanje aktivnosti glavnog korisničkog interfejsa što omogućava korisniku inicijalnu komunikaciju sa aplikacijom.

3. Opisati, navođenjem relevantnih metoda (događaja-eventa), životni ciklus aktivnosti?

- Životni ciklus aktivnosti (aktivitija) je opisan sledećim metodama:
  - **onCreate()** – poziva se prilikom prvog definisanja aktivnosti. Njome se realizuje učitavanje aktivnosti glavnog korisničkog interfejsa.
  - **onStart()** – se poziva kada aktivnost postane vidljiva korisniku. Uvek kada se pokrene aktivnost, izvršava se ova metoda.
  - **onResume()** – se poziva kada korisnik započne interakciju. Pomoću nje se startuju svi servisi ili kod koji se izvršava kada je aktivnost u prvom planu.
  - **onPause()** – se poziva kada se zaustavljaju navedeni servisi ili kod, kada aktivnost nije u prvom planu.
  - **onStop()** – se poziva kada aktivnost više nije vidljiva korisniku. Aktivnost za koju je ona pozvana prelazi u Stop stanje.
  - **onDestroy()** – se poziva pre uklanjanja aktivnosti. Njome se oslobađaju resursi.
  - **onRestart()** – nakon zaustavljanja aktivnosti, ova metoda se poziva za njeno ponovno pokretanje.

4. Navesti i objasniti načine kako se primena aktivnosti može svesti na jedan deo ekrana mobilnog uređaja?

- Aktivnost se javlja u 4 osnovna oblika:
  - aktivnost u prvom planu ima status **active** ili **running**
  - aktivnost izgubila focus a vidljiva je i dalje, znači da je **pauzirana**
  - aktivnost je potpuno u senci druge aktivnosti, znači da je **zaustavljena**
  - ako je aktivnost pauzirana ili zaustavljena, sistema može da je eliminiše iz memorije. Znači za njeno ponovno angažovanje je potrebno njeno restartovanje.

5. Šta je dijalog okvir i navesti nekoliko primera gde biste ga koristili?
  - **Dijalog okvirom** se prikazuje potvrda interakcije sa korisnikom.
6. Kako se kreira okvir dijaloga u Android aplikaciji?
  - Kreira se korišćenjem **Builder** konstruktora **AlertDialog** klase. Pri tome se vrši predefinisano zaštićene metode **onCreateDialog()**, definisane u osnovnoj klasi aktivnosti.
7. Kako se povezuju različite aktivnosti ukoliko postoje u Android aplikaciji? Navesti primer.
  - Namere (**Intens**) su mehanizam koji se koristi da bi se izvršila navigacija sa jedne aktivnosti na drugu. Kreira se objekat, koji ukazuje na aktivnost na koju se prelazi iz glavne aktivnosti.
8. Kako Android aplikacija rešava problem sudaranja namera?
  - Pozivanjem **selektora aktivnosti**.
9. Šta su Fragmenti i kako se kreiraju?
  - **Fragmenti** su detalji aktivnosti koji omogućavaju višemodularni dizajn aktivnosti. Oni se mogu posmatrati kao neka vrsta podaktivnosti.
10. Koje tipove Fragmenta poznajete?
  - Postoje:
    - Fragmenti jednog oblika (Single frame)
    - Lista fragmenata
    - Transakcije fragmenata.

11. Šta je Intent objekat i koja mu je namena?

- **Intent objekat** se kreira prilikom pozivanja aktivnosti ugrađenih aplikacija. On obezbeđuje prelazak sa jedne aktivnosti na drugu. Intent objekat ukazuje na aktivnost na koju se prelazi iz glavne aktivnosti. **Namena** mu je da omogući upotrebe aktivnosti koje su definisane u drugim aplikacijama.

12. Objasnite kako biste realizovali učitavanje ugrađenog web čitača u vašoj Android aplikaciji? Šta bi se desilo ukoliko postoje 2 čitača, na primer fabrički i Google Chrome sa istim pravom pristupa?

- Realizovao bih ga upotrebom Intent objekta sa ciljem upotrebe web čitača. Pozivao bi se selector aktivnosti, pomoću koga bi se odabrala prioriteta aktivnost za izvršenje.

13. Koji pogled omogućava prikazivanje slika u matričnoj formi sa scroll opcijom?\*\*\*

- **GridView**

14. Kako se čuvaju strukturirani podaci u Android aplikacijama?\*\*\*

- Specificirani su kao određeni URI objekti.

15. Koji tip fragmenta biste koristili prilikom kreiranja aplikacije čije izvršavanje je potrebno pauzirati usled čekanja na odgovor korisnika? \*\*\*

- **Preference Fragment**

16. Šta je neophodno uraditi da bi bilo moguće čuvanje datoteka na SD kartici mobilnog uređaja?\*\*\*

- **Kreirati odgovarajuću privilegiju** u AndroidManifest.xml datoteci.

17. Koji pogledi omogućavaju prikazivanje trenutnog vremena na ekranu?\*\*\*

- **AnalogClock** u formi analognog časovnika sa kazaljka
- **DigitalClock** u formi digitalnog časovnika.



## TEST 3

1. Šta su pogledi a šta su grupe pogleda u Android aplikacijama?
  - Element aplikacije koji ima vlastitu reprezentaciju na ekranu naziva se **pogled (Layout)**. Pogledi mogu biti grupisani u složene poglede ili **grupe pogleda (ViewGroup)**. Svaka grupa pogleda obezbeđuje način raspoređivanja komponentata korisničkog interfejsa na ekranu.
2. Navesti poznate grupe pogleda i objasnite kako se koriste?
  - **LinearLayout** – organiz. poglede u jednu vrstu ili kolonu.
  - **AbsoluteLayout** – organiz. rasp. elem. kor. interf. pomoću X i Y koordinata
  - **TableLayout** – organiz. komponente koris. interf. Po kolonama i vrstama u formi tabele.
  - **RelativeLayout** – organiz. kompon. korisn. interf. po relativnom rasporedu: gore, dole, levo, desno.
  - **FrameLayout** – izoluje deo ekrana sa ciljem prikaza pojedinačne stavke korisn. interf.
  - **ListView** – organizuje poglede u formi liste.
  - **GridView** – prikazuje stavke kor. iterf. u formi matrice.
3. Objasni ListView i GridView grupe pogleda?
  - **ListView** – organizuje poglede u formi liste.
  - **GridView** – prikazuje stavke kor. iterf. u formi matrice.
4. Koje tipove orijentacije ekrana mobilnog uređaja poznajete?
  - **Portrait** (uspravno) i **Landscape** (horizontalno).
5. Navesti i objasniti tehnike upravljanja promenom ekrana?
  - **Anchoring** – (usidrenje) ovom tehnikom se prikaz fiksira za 4 ivice ekrana
  - **Resizing** (promena veličine) i **Repositioning** (promena položaja) – je napredna tehnika promene veličine svakog pojedinačnog prikaza elementa u zavisnosti od trenutne orijentacije ekrana mobilnog uređaja.

6. Kako je moguće sačuvati podatke iz pogleda tekuće aktivnosti ukoliko se izvrši promena orijentacije ekrana?
- Da bi određene aktivnosti bile sačuvane moguće je uvek implementirati metodu **onPause()**, a zatim koristiti neke od načina za čuvanje podataka, npr: baze podataka, pomoćne datoteke, ... i sl. Mnogo jednostavniji način za čuvanje podataka neke aktivnosti je primena metode **onSaveInstanceState()**. Ona koristi Bundle objekat koji može da se iskoristi za snimanje stanja tekuće aktivnosti.
7. Šta je ActionBar i koja mu je namena?
- **ActionBar** je funkcija koja zamenjuje tradicionalnu naslovnu liniju koja se nalazila u gornjem delu ekrana. ActionBar ima namenu da prikazuje ikonu aplikacije i naziv aktivnosti.
8. Šta je neophodno dodati u AndroidManifest.xml datoteku da bi mogao da se koristi ActionBar?
- Potrebno je dodati ActionBar podršku:  
**android:theme="@style/Theme.AppCompat.Light.DarkActionBar"**
  - Prakticno, atribut android:theme, omogućava prikazivanje i uklanjanje ActionBar komponente.
9. Objasniti kako se dodaju stavke u ActionBar?
- Stavke se u ActionBar dodaju u formi menija.
10. Kada se koristi kreiranje GUI elemenata u Java klasi aktivnosti?
- Kreiranje GUI elemenata u Java klasi aktivnosti je pristup koji se koristi u situacijama kada dolazi do dinamičke promene komponenata UI u toku izvršavanja aplikacije.
11. Objasniti kako se dodaju slike za ikone i pozadinu u Android projekat? Kako se koriste u aplikaciji?
- Tako što se u folder res dodaju .png slike koje se potom koriste u aplikaciji.

12. Navedite i objasnite osnovne poglede kojima se kreira GUI Android aplikacije?

- Osnovni pogledi su:
  - TextView
  - EditText
  - Button
  - ImageButton
  - CheckBox
  - ToggleButton
  - RadioButton
  - RadioGroup

13. Objasnite kada se definiše metoda kojom se realizuje ponašanje koje je rezultat interakcije korisnika sa kontrolom (pogledom)?

- Metodom **setOnClickListener()**, omogućeno je dobijanje povratne informacije koja će biti prezentovana prilikom obraćanja pogledu. Tako je postavljen osluškivač na kontrolu, koji očekuje interakciju između korisnika i dugmeta.

14. Šta je ProgressBar i koja mu je namena?

- **ProgressBar** je kontrola koja obezbeđuje vizuelne informacije o akcijama čije izvršavanje je u toku.

15. Koje Picker poglede poznajete?

- **TimePicker** i **DatePicker**.

16. Kako biste formirali Fragment sa Picker pogledom?

- ???

17. Koje poglede za liste poznajete? Po čemu se razlikuju?

- **ListView** i **SpinnerView**.

18. Koju klasu nasleđuje klasa aktivnosti ako unapred sadrži listu?

- Klasa za implementaciju ListView pogleda nasleđuje klasu **ListActivity**.

19. Kako se eksterno mogu čuvati stavke liste u Android aplikaciji?

- U jednostavnijim aplikacijama moguće je čuvati podatke kao niz u Java klasi aktivnosti. A u većim-realnim uslovima poželjnije je čuvati stavke u bazama podataka ili u datotekama (npr: strings.xml).

20. Kako je moguće definisati broj dozvoljenih stavki za biranje iz liste u Android aplikaciji?

- Metoda **isChecked()** ispituje da li je stavka izabrana ili ne.

21. Navesti i objasniti 3 poznata specijalizovna fragmenta?

- **ListFragment** – je klasa koja definiše fragment koji sadrži ListView pogled
- **DialogFragment** – je fragment koji ima zadatak da od korisnika dobije neki odgovor, pre nego što se nastavi izvršavanje Android aplik.
- **PreferenceFragment** – je za prikazivanje aktivnosti koja korisniku omogućava opcije za prilagođavanje.

22. Za svaki od obrađenih specijalizovanih fragmenata dati po jedan primer primene?

- ListFragment, koristimo kada želimo da prikazemo listu stavki na ekranu, zajedno sa ostalim elementima korisničkog interfejsa.
- DialogFragment koristimo kada nam treba prozor sa porukom i tasterima gde će aplikacija da sačeka na odgovor korisnika.
- PreferenceFragment, se koristi kada je potreban prikaz više kategorija na ekranu a u svakoj kategoriji postoje različiti sadržaji kojima se podešavaju vrednosti u skladu sa preferencijama korisnika.

23. Sa kojim elementima upit nad provajderom sadržaja koristi formu URI identifikatora?

- Id
- Vlasnik-naziv provajdera sadržaja
- Standardni\_prefiks content://

24. Koju klasu nasleđuje klasa SMS prijemnik?

- **SmsManager**

25. Koje privilegije se dodaju u AndroidManifest datoteku za podršku slanja/prijema SMS poruka?

- android.permission.RECEIVE\_SMS
- android.permission.SEND\_SMS

26. Šta od navedenog je tačno?

- Provajdera sadržaja je neophodno uključiti AndroidManifest.xml datotekom.
- Contacts providers zahteva posebnu dozvolu za primenu.

27. Šta realizuje postavljanje kontrola na Google mapu?

- Metoda **getUISettings()** koju poziva GoogleMaps objekat.
- Objekat klase GoogleMaps.

## TEST 4

1.