

# Izazovi u realizaciji nastave na predmetu osnovnih studija Programiranje mobilnih uređaja

Marko Mićović<sup>1</sup>, Uroš Radenković<sup>2</sup>

1: Elektrotehnički fakultet Univerziteta u Beogradu

2: Elektrotehnički fakultet Univerziteta u Beogradu

elektronska pošta: [micko@etf.rs](mailto:micko@etf.rs), [uki@etf.rs](mailto:uki@etf.rs)

## REZIME

Predmet Programiranje mobilnih uređaja je izborni predmet u pretposlednjem semestru osnovnih studija odseka Softversko inženjerstvo na Elektrotehničkom fakultetu. Zbog izraženog praktičnog karaktera predmeta realizacija nastave nameće veliki broj izazova. Format nastave mora biti odgovarajući za predmet praktične prirode. Tokom čitavog semestra izučava se *Android* platforma, ali jedan semestar ne pruža dovoljno vremena da se *Android* platforma obradi u potpunosti pa stoga treba napraviti adekvatan izbor oblasti. Učestale promene vezane za programiranje mobilnih uređaja takođe otežavaju organizaciju nastave. Predstavljanje *Kotlin* programskog jezika za preporučeni programski jezik za razvoj aplikacija za *Android* platformu dodatno opterećuje gusto zbijen program predmeta. Pokretanje aplikacija na *Android* emulatoru na računarima u fakultetskoj laboratoriji može biti suviše sporo u nekim slučajevima, dok bi obezbeđivanje fizičkog mobilnog uređaja za svakog studenta predstavljalo enorman trošak. Ocenjivanje studenata mora biti pažljivo koncipirano kako se ne bi narušila praktična priroda predmeta.

**Ključne reči:** nastava, programiranje, mobilni uređaji, android, kotlin.

## 1 Uvod

Svedoci smo sveopšte prisutnosti mobilnih uređaja [1]. Pametni telefoni, satovi, televizori, automobilski multimedijalni uređaji i ostali mobilni uređaji pronalaze primenu kako u privatnim tako i u poslovnim sferama života. Krajnji potrošači koriste raznolike aplikacije za mobilne uređaje kada god se dotaknu zabave, hobija, kupovine, prevoza, sporta, zdravlja, radnih obaveza itd. Masovna upotreba mobilnih uređaja stvorila je potražnju za programerima mobilnih uređaja čiji je zadatak da naprave date aplikacije. Ovakva izražena potreba tržišta jeste jedan od razloga za formiranje predmeta koji se tiče programiranja mobilnih uređaja.

Izborni predmet Programiranje mobilnih uređaja odvija se u sedmom semestru osnovnih studija odseka Softversko inženjerstvo na Elektrotehničkom fakultetu. Izbor poslednje godine osnovnih studija, tačnije sedmog semestra, za pozicioniranje predmeta je posledica povezanosti sa drugim predmetima koji predstavljaju preduslov za uspešno savladavanje gradiva [2]. Cilj predmeta, nalik sličnim predmetima na drugim univerzitetima [3], jeste sticanje praktičnog znanja iz oblasti programiranja mobilnih uređaja. Usled vremenskog ograničenja od svega jednog semestra na predmetu se izučava samo jedna mobilna platforma da bi istu bilo moguće što detaljnije obraditi. Izabrana platforma jeste *Android* platforma zbog svoje velike zastupljenosti odnosno dominantnog udela na tržištu mobilnih uređaja [4, 5].

U nastavku rada biće navedeni najveći izazovi u realizaciji nastave na predmetu Programiranje mobilnih uređaja. Drugo poglavlje prikazuje poteškoće izazvane izabranim formatom nastave. Treće poglavlje predstavlja osvrt na odabir izučavanih oblasti programiranja mobilnih uređaja u cilju prilagođavanja predmeta za trajanje od jednog semestra. Izlaganje na temu uvođenja *Kotlin* programskog jezika u proces nastave dato je u okviru četvrtog poglavlja. Prepreke prilikom pokretanja razvijanih aplikacija u okruženju bez fizičkih uređaja u uslovima fakultetske laboratorije predstavljene su petim poglavljem. Šesto poglavlje definiše izazove i prikazuje osnovne informacije vezane za ocenjivanje studenata.

## 2 Format nastave

Programiranje mobilnih uređaja je mlada oblast trenutno u procesu intenzivnog napretka. Posledica toga jeste izostanak kvalitetnog udžbenika smatranog za standardni i ustaljeni udžbenik za nastavu u oblasti programiranja mobilnih uređaja [6]. Izraženo učestale promene, kako proširenja novim elementima tako i unapređenje ili zastarevanje postojećih, smanjuju relevantnost tradicionalnog udžbenika i otežavaju održavanje *Power Point* prezentacija za nastavu.

Praktični deo nastave održava se zato na računarima u fakultetskoj laboratoriji u vidu izrade aplikacija za mobilne uređaje. Izvorni kod aplikacije sa elementima adekvatnim za izlaganje željene tematike piše se na licu mesta zajedno sa studentima na svakom času. Ovakav model nastave iziskuje od predavača više vremena za pripremu kao i dodatan napor tokom samog trajanja časa. Vreme utrošeno svake školske godine za pripremu časova obezbeđuje usklađenost predavanih materijala sa novinama u oblasti i preporučenim praksama. Period aktivnog učestvovanja studenata u nastavi prolongiran je dodatnim naporom predavača da tokom izlaganja gradiva u svakom trenutku priđe i pomogne studentima koji naiđu na prepreku.

Tumačenje unapred pripremljenih isečaka izvornog koda na *Power Point* prezentacijama nije format nastave prilagođen za predmet praktične prirode. Diskusija o izvornom kodu aplikacije izrađene pre početka nastave takođe nije dobar pristup jer najčešće dovodi do situacije u kojoj su studenti savladani velikom količinom informacija sa kojom se odjednom susreću. Format praktičnog dela nastave opisan u prethodnom pasusu, koji podrazumeva postepeno pisanje izvornog koda na licu mesta zajedno sa studentima, je odabran upravo iz ovih razloga. Kako bi studenti kasnije tokom samostalnog učenja mogli bolje da prate tok misli predavača i da se posevete nekom od koraka u razvoju aplikacije u većoj meri za potrebe nastave koristi se softver za verzionisanje izvornog koda. Izmene načinjene tokom nastave, zaokružene u dokumentovane logičke celine, čuvaju se na fakultetskom *GitLab* serveru. Na početku školske godine za svakog studenta na predmetu pravi se korisnički nalog za fakultetski *GitLab* server kako bi kasnije mogao da pristupa materijalima.

## 3 Izučavane oblasti

Usled ograničenosti trajanja nastave na samo jedan semestar nije moguće pokriti sve grane unutar oblasti programiranja mobilnih uređaja [7]. Najveći deo nastave posvećen je odabranim temama iz zvaničnog vodiča za programere *Android* platforme [8]. Izostavljanje nekih delova pomenutog vodiča moralo je biti načinjeno zbog toga što je njegov kompletan sadržaj preobiman za jednosemestralni predmet. Tako zbog nedostatka vremena u najvećoj meri nisu pokrivene oblasti animacije i grafike. Sigurnosni aspekti razvoja aplikacija za mobilne uređaje na niskom nivou takođe se ne obrađuju zbog nedostatka vremena [9]. Teme koje se obrađuju opisane su u nastavku, a mogu se podeliti u tri celine tako da svaka celina odgovara tačno jednoj trećini semestra.

Nastava počinje obukom za korišćenje *Android Studio* razvojnog okruženja. Istovremeno objašnjava se uloga *Gradle* alata u procesu prevođenja i metod uključivanja eksternih biblioteka.

*Material Design* smernice za korisnički interfejs pregledaju se paralelno sa upoznavanjem dostupnih komponenata grafičkog korisničkog interfejsa. Izrada prve *Android* aplikacije počinje lekcijom o aktivnosti kao komponenti *Android* aplikacije, životnim ciklusima i njihovoj važnosti. Demonstrira se upotreba resursa, fragmenata i navigacione komponente iz *Jetpack* skupa biblioteka.

Druga trećina semestra počinje predstavljanjem servisa kao komponente *Android* aplikacija. Zatim sledi lekcija o tehnikama višenitnog programiranja u okviru *Android* platforme. Izučava se rad sa datotekama i lokalnim bazama podataka. Injekcija zavisnosti putem *Hilt* biblioteke je poslednje obrađivana tema u drugoj celini.

Poslednja celina obrađuje prijemnike obaveštenja i pružaoce sadržaja kao komponente *Android* aplikacije. Rad sa senzorima pokreta, pozicionim senzorima i senzorima okruženja jeste deo jednog časa. Podučava se elementarno grafičko isrcrtavanje korišćenjem `android.graphics.Canvas` klase. Rad sa kamerom i dohvaćanje mrežnih resursa u skladu sa *REST* arhitekturom korišćenjem *Retrofit* biblioteke jesu poslednje obrađivane teme na predmetu.

## 4 Programski jezik *Kotlin*

Jedna od potencijalnih poteškoća prilikom učenja programiranja mobilnih uređaja jeste nužnost predznanja iz oblasti programiranja [10]. *Android* platforma se tradicionalno oslanja na *Java* programski jezik usled čega zahteva predznanje iz oblasti objektno orijentisanog programiranja. Studenti Elektrotehničkog fakulteta izučavaju paradigmu objektno orijentisanog programiranja u okviru dva jednosemestralna predmeta što predstavlja olakšavajuću okolnost. Jedan od ta dva predmeta zasniva se upravo na *Java* programskom jeziku. Poznavanje *Java* programskog jezika u velikoj meri olakšava realizaciju nastave iz predmeta Programiranje mobilnih uređaja. Zahvaljujući predznanju studenata nastava može odmah započeti izlaganjem specifičnosti programiranja mobilnih uređaja i koncepata *Android* platforme umesto osnovama *Java* programskog jezika.

Kompanija *Google*, koja stoji iza razvoja *Android* platforme, objavila je 2017. godine da *Kotlin* postaje programski jezik prvog reda za razvoj *Android* aplikacija. Počev od tada proširenja *Android* platforme novim elementima razvijaju se imajući na umu *Kotlin* programski jezik na prvom mestu dok se za *Java* programski jezik nadalje uglavnom samo pruža podrška. *Kotlin* je danas zvanično preporučeni programski jezik za izradu aplikacija za *Android* platformu.

Prelazak na *Kotlin* programski jezik u domenu nastave na predmetu Programiranje mobilnih uređaja predstavlja ozbiljan izazov. Upoznavanje studenata sa novim programskim jezikom zahteva izvesno vreme, a oskudica vremena je već dovoljno velika nevolja uzimajući u obzir širinu oblasti izučavane na predmetu Programiranje mobilnih uređaja. Razmatrana su dva pravca delovanja radi prelaska na *Kotlin* programski jezik. Prvi proaktivniji pravac delovanja jeste usputni inkrementalni pristup upoznavanju studenata sa *Kotlin* programskim jezikom tokom trajanja čitavog semestra. Usputni inkrementalni pristup ogleda se kroz postepeno izlaganje *Kotlin* programskog jezika u vidu nezavisnih inkremenata odnosno samostalnih elemenata u trenucima kada su dati elementi apsolutno neophodni za savladavanje gradiva na predmetu Programiranje mobilnih uređaja. Ovakav pristup može biti izvodljiv po pitanju vremenskih ograničenja, ali je upitno u kojoj meri će studenti savladati *Kotlin* programski jezik i koliko bi to moglo otežati razumevanje specifičnosti programiranja mobilnih uređaja i koncepata *Android* platforme. Drugi pravac delovanja bazira se na uvođenju novog i nezavisnog predmeta na kojem se izučava *Kotlin* programski jezik. Uvođenjem kompletno novog predmeta studenti bi garantovano savladali *Kotlin* programski jezik, ali ovakav pristup otvara problem opravdanosti uvođenja takvog obaveznog predmeta. Kako se u nekoj od predstojećih školskih godina planira prelazak na *Kotlin* programski jezik neophodno je detaljnije izvršiti uporednu analizu prednosti i mana ova dva pravca delovanja.

## 5 Pokretanje aplikacija

Emulator dostupan u okviru softverskog paketa alata za razvoj *Android* aplikacija omogućava emulaciju različitih kategorija mobilnih uređaja u koje spadaju pametni telefoni, satovi, televizori i automobilski multimedijalni uređaji [11]. Nastava na predmetu bazira se primarno na radu sa pametnim telefonima kao jednim od predstavnika mobilnih uređaja. Usmerenost ka pametnim telefonima zasniva se na neoptimizovanosti ranih verzija emulatora [12]. Računari u fakultetskoj laboratoriji, gde se održava nastava, inicijalno nisu ispunjavali visoke hardverske zahteve ranih verzija emulatora čime su studenti bili prinuđeni da rade sa fizičkim uređajima. Pametni telefoni su izabrani kao fokusirana kategorija mobilnih uređaja jer većina studenata poseduje vlastiti fizički uređaj koji mogu koristiti za potrebe nastave. Pametni telefoni uglavnom pružaju i širi skup mogućnosti *Android* platforme u poređenju sa drugim kategorijama mobilnih uređaja.

Iskustveno se formirao zaključak da je studentima rad sa fizičkim uređajima svakako zanimljiviji nego rad sa emulatorom. Skup funkcionalnosti dostupan kroz emulator može uskratiti pojedine mogućnosti *Android* platforme u domenima kao što su senzori pokreta, detekcija dodira ekrana većim brojem tačaka, reprodukcija zvuka itd. Podaci dobijeni emulacijom senzora pokreta ne odgovaraju verno podacima kojima se generišu prilikom stvarnih pokreta. Emulator nema adekvatnu podršku za emulaciju dodira ekrana velikim brojem tačaka što je sputavajući faktor ukoliko aplikacija treba da detektuje broj tačaka dodira ekrana. Reprodukcija zvuka na emulatoru u uslovima fakultetske laboratorije nije moguća jer računari nemaju zvučnike.

Emulator ipak može biti prikladna zamena za fizički uređaj po pitanju bazičnih funkcionalnosti kao što je na primer rad sa komponentama grafičkog korisničkog interfejsa. Posebno treba napomenuti da je kompanija *Google* vremenom izvršila optimizaciju emulatora u velikoj meri tako da danas hardverski zahtevi emulatora ne predstavljaju problem ni za računare u fakultetskoj laboratoriji. Nezanemarljiv broj studenata upravo zato i koristi emulator na aktivnostima na kojima je to adekvatno.

## 6 Ocenjivanje studenata

Mehanizam ocenjivanja definisan je u skladu sa ciljem predmeta tako da se organizuje isključivo na računaru čime se naglašava važnost sticanja praktičnog znanja. Ocenjivanje se sprovodi kroz veći broj praktičnih aktivnosti, da bi studenti bili motivisani na kontinuirani rad, navedenih u tabeli 1. Aktivnosti za proveru znanja obuhvataju tri laboratorijske vežbe, dva kolokvijuma, ispit i odbranu projekta odnosno domaćeg zadatka. Prilikom svake provere znanja pred studenta se postavlja zadatak da napravi potpuno funkcionalnu aplikaciju koja predstavlja zaokruženu celinu. Ocenjivanje svih aktivnosti za proveru znanja vrši se uvidom u rešenje tokom same aktivnosti kada se ujedno obavlja i usmeno ispitivanje studenata. Nenultim brojem poena ocenjuju se samo oni delovi rešenja koji su potpuno funkcionalni čime su studenti primorani da se samostalno uhvate u koštac sa grešakama u aplikaciji. Pobuda za ovakav način ocenjivanja jeste želja da se kod studenata eliminiše strah od procesa otklanjanja grešaka.

Laboratorijske vežbe spadaju u grupu najlakših aktivnosti sa zadacima čija su rešenja najmanjeg obima. Vreme za izradu rešenja ograničeno je na dva sata. Laboratorijske vežbe su zamišljene sa namerom da podstaknu studente na što ranije moguću pripremu naredne teže aktivnosti. Tako prva laboratorijska vežba predstavlja aktivnost na kojoj se zahteva poznavanje komponenti grafičkog korisničkog interfejsa, aktivnosti kao komponente *Android* aplikacija i životnih ciklusa što je samo deo gradiva koji se kasnije proverava prvim kolokvijumom. Druga laboratorijska vežba pokriva oblasti servisa kao komponente *Android* aplikacija i višenitnog procesiranja na mobilnim uređajima koje su neophodne pri izradi drugog kolokvijuma. Poslednja odnosno treća laboratorijska vežba

predstavlja pripremu za ispit, a tiče se oblasti prijemnika obaveštenja kao komponente *Android* aplikacije, senzora i grafičkog iscrtavanja.

Kolokvijumi spadaju u grupu srednje teških aktivnosti. Vreme za izradu rešenja ograničeno je na tri sata. Gradivo obuhvaćeno kolokvijumima oslanja se direktno na prethodeće laboratorijske vežbe. Prvi kolokvijum pored gradiva sa prve laboratorijske vežbe proverava i poznavanje resursa, fragmenata i navigacije među njima. Drugi kolokvijum proširuje skup oblasti pokrivenih na drugoj laboratorijskoj vežbi radom sa datotekama, lokalnom bazom podataka i injekcijom zavisnosti putem *Hilt* biblioteke.

U okviru ispitnog organizuju se poslednje dve aktivnosti: ispit i odbrana projekta. Ispit spada u najtežu aktivnost sa zadacima čija su rešenja najvećeg obima. Vreme za izradu rešenja ograničeno je iz formalnih razloga na tri sata, ali studentima je dopušteno da nastave sa radom i nakon isteka tog roka. Prosečno vreme izrade rešenja je između četiri i pet sati. Ispit obuhvata celokupno gradivo predmeta. Pored gradiva obuhvaćenog svim prethodno navedenim aktivnostima ispit uključuje i pružaocce sadržaja kao komponentu *Android* aplikacije, rad sa kamerom i pristup mrežnim resursima. Aplikacija koju student treba da napravi na ispitu može sadržati sve elemente predavane tokom semestra. Odbrana projekta realizovana je kao modifikacija domaćeg zadatka. Uspešnom izradom modifikacije predatog projekta studenti potvrđuju da im je projekat poznat odnosno da su isti radili samostalno. Projektovano vreme za izradu rešenja projekta jeste sedam do četrnaest dana. Projekat, nalik ispitu, obuhvata celokupno gradivo predmeta ali se od ispita razlikuje po obimu. Aplikacija koju student pravi u okviru projekta sadrži značajno veći broj funkcionalnosti nego što je to slučaj sa ispitom.

Tabela 1, Pregled aktivnosti za proveru znanja

Naziv	Vreme održavanja	Udeo u konačnoj oceni	Glavne karakteristike uobičajeno zadatih aplikacija
Prva lab. vežba	4. nedelja semestra	3%	Jednostavna aplikacija sa aktivnostima, elementarnim funkcionalnostima, reagovanjem na promenu konfiguracije i bez perzistencije podataka.
Prvi kolokvijum	5. nedelja semestra	10%	Aplikacija sa jednom aktivnošću, fragmentima, navigacijom među fragmentima, reagovanjem na promenu konfiguracije i bez perzistencije podataka.
Druga lab. vežba	9. nedelja semestra	3%	Jednostavna aplikacija sa servisima i potrebom za višenitnim procesiranjem.
Drugi kolokvijum	10. nedelja semestra	10%	Aplikacija za rad sa datotekama i mogućnošću perzistencije podataka u lokalnoj bazi podataka.
Treća lab. vežba	14. nedelja semestra	4%	Jednostavna aplikacija za obradu i iscrtavanje podataka dobijenih sa senzora.
Ispit	Ispitni rok	20%	Složena aplikacija koja može sadržati bilo koji element predavan tokom nastave.
Projekat	Ispitni rok	50%	Složena aplikacija sa svim predavanim elementima i velikim brojem funkcionalnosti.

Prosečan uspeh studenata na skali od 0 do 100 poena u prethodne tri školske godine prikazan je u okviru slike 1 za svaku od aktivnosti za proveru znanja. Može se primetiti trend nešto slabijeg uspeha studenata na drugoj laboratorijskoj vežbi i drugom kolokvijumu pošto se ove aktivnosti za proveru znanja održavaju sredinom semestra kada su studenti generalno najopterećeniji. Najbolji uspeh studenti generalno ostvaruju na projektu odnosno domaćem zadatku jer imaju dovoljno vremena za njegovu izradu i mogu se posvetiti testiranju.



Slika 1 Prosečan uspeh studenata na skali od 0 do 100 poena na aktivnostima za proveru znanja

Prema mišljenju studenata predmet Programiranje mobilnih uređaja spada u grupu subjektivno težih izbornih predmeta jer je za uspešno savladavanje gradiva potrebno više vremena u poređenju sa drugim izbornim predmetima. Kategorizacija među teže izborne predmete je očekivana pošto predmet predstavlja spoj koncepata naučenih na prethodnim godinama studija i *Android* platforme. Ipak i pored toga studenti predmet smatraju interesantnim jer format nastave stavlja akcenat na praktičan rad kao i zbog toga što je znanje iz ove oblasti danas primenljivo. Krajnji ishod predmeta jeste da je student nakon uspešno položenog ispita osposobljen za samostalnu izradu potpuno funkcionalnih aplikacija za *Android* platformu.

## 7 Zaključak

U okviru rada izloženi su najveći izazovi u realizaciji nastave na predmetu Programiranje mobilnih uređaja. Predstavljani su uzroci datih izazova kao i strategije kojima oni mogu biti prevaziđeni. Ustanovljeno je da nastava na računarima, u vidu izrade aplikacije za mobilne uređaje zajedno sa studentima, predstavlja najbolji format nastave u uslovima stalnih promena *Android* platforme, izostanku tradicionalnog udžbenika i praktične prirode predmeta. Ovakav format nastave sa druge strane iziskuje dodatan napor od strane predavača. Prilikom odabira oblasti izučavanih na predmetu morale su biti izostavljene neke od tema vodiča za programere *Android* platforme. Objavlivanjem odluke da *Kotlin* postaje programski jezik prvog reda za razvoj *Android* aplikacija javlja se potreba za njegovim uvođenjem u nastavu na predmetu. Izazov uvođenja *Kotlin* programskog jezika još nije prevaziđen ali razmatraju se dva moguća pravca za njegovo prevazilaženje. Prikazan je pregled poteškoća pri pokretanju aplikacija u uslovima fakultetske laboratorije na koje se nailazilo tokom više godina držanja nastave na predmetu. Mehanizam ocenjivanja sa velikim brojem aktivnosti tokom semestra definisan je sa ciljem da se prevaziđe izazov kvalitetnog ocenjivanja studenata i motivacija studenata na praktičan rad.

Studenti iz oblasti programiranja mobilnih uređaja neretko pišu diplomske radove jer se predmet održava u pretposlednjem semestru osnovnih studija i pokriva interesantnu oblast. Tokom poslednje tri školske godine u proseku petina studenata, koji su pratili nastavu iz predmeta Programiranje

mobilnih uređaja, svoje diplomske radove piše na neku temu iz ove oblasti. U sklopu diplomskih radova studenti najčešće realizuju aplikacije zabavnog karaktera, kao što su video igre, pošto im je to najzanimljivije.

Tokom nastave u narednim školskim godinama biće skrenuta pažnja na različite kategorije aplikacija nezabavnog karaktera pošto se trenutno studenti masovno fokusiraju na izradu aplikacija zabavnog karaktera. U nekoj od narednih školskih godina planira se i prelazak na *Kotlin* programski jezik koji je preporučeni programski jezik za razvoj *Android* aplikacija. Prelazak na *Kotlin* programski jezik biće veliki izazov jer u postojeći fond časova treba dodati koncepte programskog jezika *Kotlin* sa kojim se studenti nisu susretali na ranijim godinama studija.

## Literatura

- [1] Oberlo. 10 Mobile Usage Statistics Every Marketer Should Know in 2021 [Infographic] [Internet]. 2021 [cited 2021 Oct 3]. Available from: <https://www.oberlo.com/blog/mobile-usage-statistics>
- [2] Qusay M, Popowicz P. A Mobile Application Development Approach to Teaching Introductory Programming. IEEE Frontiers in Education Conference (FIE). 2010 Oct 27. doi: [10.1109/FIE.2010.5673608](https://doi.org/10.1109/FIE.2010.5673608)
- [3] Ekong DU. A Survey of Android Programming Courses. Proceedings of IEEE Southeastcon. 2012 Mar 15. doi: [10.1109/SECon.2012.6196987](https://doi.org/10.1109/SECon.2012.6196987)
- [4] Wei H, Hong G. Curriculum Architecture Construction of Mobile Application Development. International Symposium on Information Technologies in Medicine and Education. 2012 Aug 3. doi: [10.1109/ITiME.2012.6291243](https://doi.org/10.1109/ITiME.2012.6291243)
- [5] Statcounter Global Stats. Global Operating System Market Share Worldwide [Internet]. 2021 [cited 2021 Sep 7]. Available from: <https://gs.statcounter.com/os-market-share/mobile/worldwide>
- [6] Burton BG, Martin BN. Six Years of Teaching Mobile Computing: What We Have Learned. IEEE 16<sup>th</sup> International Conference on Advanced Learning Technologies (ICALT). 2016 Jul 25. doi: [10.1109/ICALT.2016.112](https://doi.org/10.1109/ICALT.2016.112)
- [7] Ma Y, Mei J, Xiao Y, Weilong L. Teaching Content, Case Design and Evaluation System of Mobile Devices Programming Course. 9<sup>th</sup> International Conference on Information Technology in Medicine and Education (ITME). 2018 Oct 19. doi: [10.1109/ITME.2018.00143](https://doi.org/10.1109/ITME.2018.00143)
- [8] Android for Developers. Developer Guides [Internet]. 2021 [cited 2021 Sep 7]. Available from: <https://developer.android.com/guide>
- [9] Xiaohong Y, Williams K, McCrickard S, Hardnett C, Lineberry LH, Bryant K et al. IEEE Frontiers in Education Conference (FIE). 2016 Oct 12. doi: [10.1109/FIE.2016.7757365](https://doi.org/10.1109/FIE.2016.7757365)
- [10] Tigrek S, Obadat M. Teaching Smartphones programming using (Android Java): Pedagogy and Innovation. International Conference on Information Technology Based Higher Education and Training (ITHET). 2012 Jun 21. doi: [10.1109/ITHET.2012.6246039](https://doi.org/10.1109/ITHET.2012.6246039)
- [11] Android for Developers. Run Apps on the Android Emulator [Internet]. 2021 [cited 2021 Sep 7]. Available from: <https://developer.android.com/studio/run/emulator>
- [12] JoliCode. Speed up your Android Emulator [Internet]. 2013 [cited 2021 Sep 7]. Available from: <https://jolicode.com/blog/speed-up-your-android-emulator>