**KAUNO TECHNOLOGIJOS UNIVERSITETAS**

**INFORMATIKOS FAKULTETAS**

**Iš medžio iškritę**

**Programų kūrimas „Android“ aplinkai**

**Komandinis darbas**

# Priėmė:

# Dėst. M. Binkis

# Dėst V. Keršienė

# Atliko:

# IFF-4/6 D. Vaitkevičius

# IFF-4/6 P. Maldonis

# IFF-4/6 E. Bureika

# IFIN-3/4K. Balčiūnas

**KAUNAS, 2024**

**Turinys**

[1. Įvadas 3](#_Toc185153536)

[2. Programų kūrimas „Android“ aplinkai 3](#_Toc185153537)

[1. Įvadas 3](#_Toc185153538)

[2. Temos analizė ir detalus aprašymas 4](#_Toc185153539)

[3. Pagrindiniai kūrimo įrankiai ir priemonės 5](#_Toc185153540)

[4. Esama padėtis 6](#_Toc185153541)

[5. Panaudojimo pavyzdžiai 7](#_Toc185153542)

[6. Programinių kalbų apžvalga „Android” programų kūrimui 9](#_Toc185153543)

[9](#_Toc185153544)

[7. Programėlės kūrimas 10](#_Toc185153545)

[8. Įdomūs faktai, susiję su programų kūrimu „Android“ 11](#_Toc185153546)

[3. Išvados 12](#_Toc185153547)

[4. Literatūra 13](#_Toc185153548)

1. Įvadas

Šio projekto dalyviai - KTU informatikos fakulteto studentai. „Android“ programėlių kūrimas - mūsų studijų krypčiai artima tema, o ir pati „Android“ sistema – viena populiariausių pasaulyje. Nors projekto dalyviai yra iš skirtingu studijų modulių, bet bendras tikslas supažindinti apie Android sistema ir jai skirtų programėlių kūrimą yra vienodas. Todėl šiame referate siekiama išnagrinėti „Android“ programinės įrangos kūrimo ypatumus, apžvelgti įvairius įrankius, naudojamus programų kūrimui, jų populiarumą bei funkcionalumą. Taip pat – pateikti įvairių pavyzdžių, įžvelgti galimas ateities perspektyvas. Pagrindinis tikslas - supažindinti paprastus „Android“ vartotojus su jų vartojamų programėlių sukūrimu.

1. Programų kūrimas „Android“ aplinkai
2. Įvadas

Paveikslėlis, kuriame yra tekstas, ekrano kopija, Šriftas, diagrama

Automatiškai sugeneruotas aprašymas„Android“ kompanija susikūrė 2003 metais, o pirmoji jos versija buvo išleista 2008 metais. Nuo to laiko ši programinė įranga patobulėjo, jos funkcionalumas žymiai pasikeitė, ir dabar tai yra viena iš populiariausių operacinių sistemų, naudojamų milijarduose mobiliųjų įrenginių visame pasaulyje. Šiomis dienomis pasaulyje yra apie 3.3 milijardo „Android“ vartotojų. Prie šios sistemos populiarumo prisideda didelis kiekis šiai sistemai pritaikytų programėlių, o vartotojų poreikis turėti naujas ar atnaujintas programėles nuolat augs, todėl ši tema bus aktuali ilgą laiką. 2018 metais buvo atliktas tyrimas, kuriame buvo palyginta Kinijos bei „Google Play“ rinkos.[1] Šis tyrimas parodė, kad visose rinkose yra virš 6 milijonų programėlių bei apie milijonas programinės įrangos kūrėjų. Tokiam plačiam vartotojų poreikiui patenkinti yra daug programinės įrangos kūrėjų. Android operacinės sistemos populiarumui nekrentant ir vartotojų skaičiui išliekant stabiliam, programėlių poreikis išlieka taip pat stabilus. Artimiausias konkurentas „Apple”, kuris turi apie 30% rinkos, net nepriartėja prie to, kad valdo „Google” su savo „Android” sistema, kuri užima net 70% pasaulinės rinkos. Per laiką buvo nedidelių šuolių, bet konkurentams sunku pasiekti pusę rinkos. Kalbant apie mažesnes operacines sistemas, jos visiškai yra užgožtos ir kol kas neturi jokių šansų pasiekti, net „Apple”. Nors praeityje tokios sistemos kaip „Microsoft phone”, kurios bandė užimti dalį rinkos, bet dėl savo trūkumų, tokių kaip pirmoje versijoje „Windows 7”, kur nebuvo galima laisvai parsisiųsti failų iš interneto ir per mažo naudojimo telefonuose ši operacinė sistema netrukus žlugo. Tuo tarpu artimiausias konkurentas „Apple” negali užimti tokios pat rinkos, bet jų tikslas yra pateikti save kaip prabangos ženklą, dėl to jų telefonai nėra nutaikyti į pigesnius segmentus ir tai galėtų būti viena iš priežasčių, dėl ko jiems nepavyksta užimti didesnės dalies rinkos. Nors operacinė sistema pasižymi gera sparta, nemažu funkcionalumu, bet iki šiol buvo ribojama naudoti internete instaliacinių failų, teigiant, kad tai daroma dėl saugumo, bet vėliau tai pasikeitė. Kitas aspektas - norint programuoti „IOs” sistemai reikia įvykdyti griežtesnius reikalavimus ir tenka mokėti 99 dolerių mokestį kas metus, kad taptum registruotu programuotoju, kai tuo tarpu „Android” sistemai tai tėra 25 doleriai ir tai yra vienkartinis mokestis, dėl to šioje platformoje dažnai verta turėtu nemokamas programėles, kurios gali uždirbti iš reklamų. Bet tuo pačiu reikia būti atsargiam, nes didelis reklamų naudojimas gali atbaidyti vartotojus. Internetinėje erdvėje esant kovai tarp „Apple IOs” ir „Google Android” operacinių sistemų dažnai pabrėžiama, kad „Android” yra stringanti, nesaugi operacinė sistema. Kai tai iš dalies buvo tiesa prieš gerą dešimtmetį, šiuo metu „Android” yra pasikeitusi ir jos našumas bei saugumas yra kur kas patobulėjęs. Strigimo problemos buvo ištaisytos optimizuojant sistemą, su kiekviena iteracija, buvo krypstama geresne vartotojui linkme. Nors didesnė dalis kaltės tenka procesorių gamintojams, kurie dažnai anksti nutraukdavo palaikymą ir dėl to atnaujinimai turėdavo būti nutraukti. Šiuo metu šita situacija yra pasikeitusi ir atnaujinimai vykdomi žymiai ilgesnį laiko tarpą. Programavimo atžvilgiu yra labai daug šaltinių, kurie naujam programėlių kūrėjui gali padėti, o pats „Android” turi gerai aprašyta dokumentacija, daug kodo pavyzdžių ir norint sukurti nesudėtingą programėlę, tą padaryti yra tikrai nesudėtinga.

1. Temos analizė ir detalus aprašymas

Programos kūrimas „Android“ aplinkai susideda iš daug skirtingų aspektų. Šiais laikais tą atlikti yra gan paprasta, dėl didelio kiekio egzistuojančių šablonų bei kūrimui skirtų įrankių bei daugybės pavyzdžių bei pamokų, kaip tą padaryti. Ivano Trogličiaus [straipsnyje](https://decode.agency/article/android-app-development/) apie „Android“ programėlių kūrimą [3] yra pateikti keli žingsniai, kaip sukurti programėlę. Pats pirmas etapas - idėjos generavimas bei projekto planavimas. Prieš kuriant programėlę reikia nustatyti programėlės paskirtį - kokią problemą padės spręsti ši programėlė. Tam reikia atlikti rinkos analizę, paieškoti jau sukurtų panašaus tipo programėlių bei pasidomėti sunkumais, kurie gali kilti kuriant programėlę. Tai gali padėti išvengti nesėkmių bei užkirsti kelią problemoms kuriant programėlę. Jei programėlė kuriama rinkai, reikia apgalvoti jos kūrimo bei palaikymo finansavimą. Galimi keli finansavimo tipai:

1. Mokama programėlė - kai vartotojas turi sumokėti vienkartinį mokestį, kad galėtų naudotis programėle.
2. Prenumeruojama programėlė - kai vartotojas turi susimokėti mokestį kas nustatytą laiko tarpą, kad galėtų naudoti programėle ar papildomomis jos funkcijomis.
3. Programėlė su integruotomis reklamomis – tai vienas populiariausių būdų šiomis dienomis, nes jis leidžia vartotojams nemokamai atsisiųsti ir naudotis programėle, tačiau joje yra integruotų reklamų.

Tolimesni programėlės kūrimo žingsniai yra susiję su technologijų pasirinkimu - programavimo kalba, IDE[[1]](#footnote-1), bei programos karkasą. Išsirinkus įrankius reiktų vizualizuoti programėlės karkasą bei numatomą vartotojo dizainą. Tai gali palengvinti bei pagreitinti programėlės kūrimo eigą. Kitas žingsnis - visko apjungimas bei idėjos įgyvendinimas. Šiame etape yra rašomas visas kodas, kuriama vartotojo sąsaja, programėlės veikimo algoritmai bei sąsaja su pagrindiniais serveriais. Tai – svarbiausia bei daugiausiai laiko užimanti programėlės kūrimo proceso dalis, nes ji nulemia, kaip greitai veiks pati programa, ar ji veiks tinkamai. Sukūrus programą, būtina ją pratestuoti tiek automatiniais, tiek rankiniais testais. Viskam pavykus, galima publikuoti programėlę „Google Play“ platformoje. Reikia nepamiršti nuodugniai patikrinti, ar programėlė nepažeidžia šios platformos nuostatų bei taisyklių bei paruošti programėlę APK[[2]](#footnote-2) failų rinkiniu.

Paveikslėlis, kuriame yra tekstas, ekrano kopija, multimedija, Operacinė sistema

Automatiškai sugeneruotas aprašymas

pav. Pagrindiniai programėlės kūrimo etapai

Luko Moisiejaus [7] magistro baigiamojo darbo „Programėlių išmaniesiems įrenginiams kūrimo technologijų analizė“ išvadose yra pabrėžiama, kad tobulėjantys interaktyvūs programėlių kūrimo įrankiai leidžia jas kurti net neturint programavimo patirties - „ Tyrimo metu išbandyti du interaktyvūs vizualaus kūrimo įrankiai, išsiaiškintos jų galimybės, bei sukurtos dvi pilnai funkcionuojančios mobiliosios programėlės. Tuo įrodyta, kad savo jėgomis, neturint programavimo žinių ir įgūdžių, galima įgyvendinti savo idėją ir sukurti mobiliąją programėlę.“

1. Pagrindiniai kūrimo įrankiai ir priemonės

Pagrindinės Android programinės įrangos kūrime naudojamos programavimo kalbos – Java ir Kotlin. Java – viena iš populiariausių programavimo kalbų šiomis dienomis, turinti platų panaudojimo spektrą - nuo kompiuterinių programų iki programėlių išmaniuosiuose telefonuose. Arvydo Putnos magistro baigiamajame darbe pateiktame C# ir Java programavimo kalbų analizėje [5] pateikiamos Java kalbos savybės. „Pagrindinis šios kalbos tikslas - kurti programinę įrangą įvairiausiems elektroniniams įrenginiams, nepriklausomai nuo naudojamų procesorių tipų.“ Kad galima būtų tai įgyvendinti, ši kalba parašytas kodas negeneruojamas tiesiai procesoriui, o yra kompiliuojamas į baitkodą, kuris gali būti paleidžiamas bet kokioje Javos virtualiojoje mašinoje (JVM), nepriklausomai nuo kompiuterio. Būtent dėl to ši kalba taip plačiai paplitusi tarp Android programinės įrangos kūrėjų. Štai dar keletas Java programavimo kalbos privalumų:

* Paprastumas – Java yra greitai išmokstama kalba, bei paveldėjo daugumą sintaksės iš C kalbos grupių, todėl turint programavimo patirties su C ar C++ kalbomis, perprasti Java nebus sudėtinga.
* Saugumas – Java užtikrina saugumą, nes jos pagrindu parašytos programos yra vykdomos Java aplinkoje ir neleidžia pasiekti kompiuterių resursų.
* Objektiškumas - Java yra objektinio programavimo pagrindu sudaryta kalba.
* Populiarumas – Java kalba naudojasi daug programinės įrangos kūrėjų.

Kita „Android“ programėlių kūrimui naudojama kalba – Kotlin. Nuo 2018 metų ši kalba yra oficiali „Android“ aplikacijų kūrimo kalba. Moksliniame žurnale „International Journal Of Advanced Research In Computer Science“ [6] buvo aptarta Kotlin programavimo kalba bei jos privalumai, lyginant su Java:

* Kotlin kalbos galimybės leidžia programuotojams efektyviau kurti programėles.
* Kotlin kompiliatorius daro daug programos patikrinimų, kad išvengtų klaidų, tai daro šią kalbą viena iš saugiausių lyginant su kitomis.
* Šios kalbos produktyvumas – kodo ilgis lyginant su Java yra kur kas trumpesnis.

Paveikslėlis, kuriame yra tekstas, ekrano kopija, Šriftas

Automatiškai sugeneruotas aprašymas

pav. Java ir Kotlin programavimo kalbų palyginimai

Kuriant „Android“ programėles dažnai naudojamas XML[[3]](#footnote-3)*.* Tai yra bendrosios paskirties duomenų struktūrų bei jų turinio aprašomoji kalba. Pagrindinė šios kalbos paskirtis - užtikrinti lengvą duomenų keitimąsi tarp skirtingų tipų sistemų. Būtent todėl ši kalba taip dažnai naudojama tarp „Android“ programinės įrangos kūrėjų, nes šią operacinę sistemą palaikančių įrenginių įvairovė - labai didelė. XML „Android“ aplikacijų kūrime naudojamas naudotojo sąsajai kurti, apibrėžti išteklius, tokius kaip spalvos, stiliai, tekstiniai failai ar matmenys, meniu struktūros apibrėžimui.

1. Esama padėtis

Android operacinė sistema yra viena populiariausių visame pasaulyje, užimanti apie 70% viso pasaulio mobiliosios rinkos ir aptarnaujanti daugiau nei 3.3 milijardo vartotojų. Tai suteikia galimybę „Android“ sistemai tapti pagrindiniu įrankiu kuriant programas, kurios pasiekia platesnį vartotojų ratą. „Android“ ekonominė sistema ir vis dar auga ir tikrai dar augs, nes kasmet atsiranda tūkstančiai naujų, patobulėjusių ir pagerėjusių programų, kurias vysto ir kuria kūrėjai iš viso pasaulio. Oficialiame „Android“ kūrėjų puslapyje [8] galima pamatyti bei palyginti su kitomis platformomis, tokiomis kaip iOS. „Android“ išlieka prieinamesnė, nes reikalauja mažesnių finansinių barjerų kūrėjams.

|  |  |
| --- | --- |
| **Platforma** | **Kūrėjo paskyros registracija** |
| Google Play | 25 EUR |
| Apple iOS | 99 EUR |

Android sistema pasižymi dideliu programėlių asortimentu, kas leidžia naudotojams ir kūrėjams prisijungti prie įvairesnių grupių. Ši sistema apima ne tik išmaniuosius telefonus ir planšetinius kompiuterius, bet ir išmaniuosius televizorius, dėvimus įrenginius bei automobilių sistemas. Dėl šios priežasties „Android“ kūrimas tampa ypač patrauklus kūrėjams, norintiems pasiekti kuo platesnį vartotojų ratą.

Nuo pat pirmųjų metų bei jos versijų „Android“ sistema vis tobulėja su kiekviena diena, o naujos versijos įtraukia tiek našumo gerinimus, tiek naujas funkcijas, kurios leidžia kurti vis modernesnes ir pažangesnes programas. Per pastaruosius metus „Android“ tapo daug saugesnė platforma kuri suteikia pasitikėjimo kūrėjams, o įvairūs patobulinimai užtikrina, kad programėlės veiks patikimai ir greitai net ir senesniuose įrenginiuose.

* Šiuolaikiniai kūrimo įrankiai ir technologijos

Išmaniosioms technologijoms ir programėlėms, skirtoms Android programėlėms kurti, išsiskiria pagrindinės programavimo kalbos, tokios kaip Java ir Kotlin. Java kaba jau ilgus metus buvo pagrindinė „Android“ kūrimo kalba dėl savo universalumo ir suderinamumo su įvairiomis platformomis. Tačiau nuo 2018 metų Kotlin kalba tapo oficialia ir pagrindine kalba „Android“ programėlių kūrimui ir vystymui ir dabar vis dažniau naudojama dėl savo paprastos saugumo funkcijos, sintaksės ir greitesnio programų proceso kūrimo etapuose [9]. Kotlin leidžia programuotojams bei kūrėjams rašyti trumpesnį ir efektyvesnį kodą, sumažinant kodo ir kūrimo klaidų , kas ypatingai svarbu kuriant kokybiškas ir našias programas bei .

Android Studio yra oficiali integruota plėtros aplinka (IDE), kuri padeda kūrėjams kurti, testuoti ir optimizuoti „Android“ programėles bei daugiau jų gaminių. Šis įrankis teikia visus reikalingus įrankius, tokius kaip automatiniai klaidų tikrinimo įrankiai, testavimo įrankiai ir analizės priemonės, leidžiančios geriau suprasti, kaip programa veikia ir kaip ją patobulinti [10].

* Programų kūrimo metodai ir tendencijos

Šiuo metu Android vystymosi ir kūrimo ekosistemoje yra daugybė technologinių pokyčių, ypač dirbtinio intelekto ir VR[[4]](#footnote-4) srityse. Programos, integruojančios dirbtinį intelektą ir mašininį mokymąsi, leidžia kurti personalizuotas vartotojo patirtis, kuriose programos gali adaptuotis pagal vartotojo elgesį ir poreikius. Labai geras pavyzdys, programos, naudojančios „Google ML Kit“ ar „TensorFlow“, puikiai suteikia galimybę kūrėjams įdiegti pažangias AI[[5]](#footnote-5) funkcijas net ir neturint pakankamai dirbtinio intelekto žinių [11]. Taip pat, papildyta realybė kuri kuo toliau, tuo labiau tampa populiaresnė, nes leidžia kurti interaktyvias ir įtraukiančias programas, kurios gali būti naudojamos tiek žaidimuose, tiek edukacijoje, tiek medicinoje [12]. Dėka šių pažangių ir populiarių technologijų Android sistema tapo dar patrauklesnė kūrėjams, norintiems pasiūlyti inovatyvias ir unikalias vartotojo patirtis.

* Android programų kūrimo iššūkiai

Vienas iš pagrindinių problemų bei iššūkių, su kuriais susiduria „Android“ kūrėjai, yra įrenginių ir programinės įrangos įvairovė. Taigi, „Android“ platforma turi puikia galimybę veikti įvairiuose įrenginiuose, nuo naujausios klasės telefonų iki mažesnių, senesnių bei pigesnių įrenginių, tačiau dažnai tenka susidurti su suderinamumo problemomis ir optimizavimo iššūkiais. Kūrėjai turi užtikrinti, kad jų programos veiktų ko puikiausi visuose įrenginiuose, nes skirtingi įrenginiai gali turėti skirtingus procesorius, ekrano dydžius ir operacinės sistemos versijas. Tačiau „Google“ reguliariai teikia atnaujinimus, kurie padeda geriau tvarkyti šiuos iššūkius [13]. Bet realybė dažnai gali būti visai kitokia. Java kalba, kuri naudojama „Android” programėlių kūrimui ir teoriškai ta pati kalba ir sistema galima naudoti daugelyje įrenginių, bet programuotojai netruko Java kalbos šūkį “Write one run everywhere”[[6]](#footnote-6) pakeistį šūkiu “Write once debug everywhere[27]”[[7]](#footnote-7). Taip nutiko dėl to, kad dažnu atveju skirtingos platformos neįjungia reikalingų servisų ar tie servisai yra prijungti skirtingai.

* Ateities perspektyvos

Ateityje ir toliau bėgant metams „Android“ sistema ir toliau tobulins kūrimo galimybes, pasitelkdama ir prijungdama naujas technologijas, tokias kaip 5G internetas, kurios leis kurti dar spartesnes ir interaktyvesnes programas. Taip pat, toliau bus didinamas dėmesys programų saugumui ir naudotojo privatumo apsaugai, kad būtų užtikrinta, jog vartotojų duomenys būtų apsaugoti nuo kibernetinių grėsmių [11]. Programų kūrėjai ir toliau turės galimybę pasinaudoti moderniausiais įrankiais ir kalbomis, tokiomis kaip pavyzdžiui, Kotlin, kad sukurtų dar novatoriškesnes programas.

1. Paveikslėlis, kuriame yra tekstas, įtaisas, ekrano kopija, Nešiojamas ryšio įrenginys

   Automatiškai sugeneruotas aprašymasPanaudojimo pavyzdžiai

„Android“ operacinė sistema virto vienu iš pagrindinių įrankių ir priemonių įvairių tipų programų kūrimui, kurios naudojamos beveik kiekvienoje mūsų kasdieninės veiklos srityje. Dėl savo universalumo ir patogumo, galimybės naudoti technologijas įvairiose industrijose, „Android“ suteikia kūrėjams platų galimybių spektrą, kaip pasiekti vartotojus ir patenkinti įvairius poreikius. Žemiau pateikiami keli svarbiausi „Android“ technologijų panaudojimo pavyzdžiai:

* Socialiniai tinklai ir mobilioji komunikacija

Socialiniai tinklai ir mobilioji komunikacija apima ir užima vieną iš pagrindinių vietų „Android“ programų pasaulinėje rinkoje. Tokių programų kaip „Instagram“, „Facebook“, „WhatsApp“ ir daugelio kitų vystymasis bei jų tobulinimas yra tiesiogiai susijęs su šios operacinės sistemos technologinėmis galimybėmis ir poreikiais. Šios platformos pasiekė didžiulį vartotojų ratą ir tapo neatsiejama tiek kasdieninio gyvenimo, tiek verslo komunikacijos dalimi. Pavyzdžiui, „Instagram“ leidžia žmonėms lengvai pasidalyti savo nuotraukomis ir vaizdo įrašais, tuo pačiu integruojant socialinius tinklus ir reklamos galimybes, o šios funkcijos tapo itin svarbios tiek mažoms įmonėms, tiek prekės ženklams, norintiems pasiekti auditoriją [14].

* Sveikatos technologijos ir asmeninė priežiūra

Mobiliosios programos taip pat labai prisideda prie svarbiausios žmogui vertybės tai sveikatos technologijų ir asmeninės priežiūros srityje. „Android“ suteikia puikią aplinką programoms, tokioms kaip „MyFitnessPal“, kurios stebi vartotojo sveiką gyvenseną bei mitybą, aktyvumą ir sveikatos duomenis. Pasitelkus išmaniuosius telefonus, papildomus įrenginius ir jų jutikliais, šios programos gali kuo puikiausiai ir tiksliai sekti vartotojo kūno būklę ir duoti gerų patarimų, kaip pagerinti gyvenimo kokybę ir laiką. Taip pat, didelė dalis šių programų taip pat integruojasi su dėvėjimo įrenginiais, tokiais kaip išmanieji laikrodžiai, kurie naudoja „Android“ operacinę sistemą, taip suteikdami papildomų duomenų ir įžvalgų apie vartotojo sveikatą ir aktyvumą [15].

* Išmanieji namai ir jų automatizavimas

„Android“ taip pat tapo svarbia išmaniųjų namų technologijų dalimi. Programos, tokios kaip „Google Home“ ir „Amazon Alexa“, leidžia vartotojams valdyti savo namų įrenginius, tokius kaip išmanias lempas, šildymo ar saugumo sistemas ir kt., tiesiogiai per mobiliąją programą. „Android“ ekosistema suteikia galimybę naudoti balso komandas ir automatizuoti kasdienius veiksmus, tokius kaip temperatūros reguliavimas ar šviesos įjungimas ar išjungimas, taip padidinant patogumą ir saugumą. „Google Home“ pavyzdys rodo, kaip „Android“ technologija prisideda prie išmaniųjų namų pritaikymo, padedant vartotojams geriau kontroliuoti savo aplinką [16].

* Papildyta realybė ir žaidimai

Papildyta realybė (AR) ir žaidimų pramonė yra viena iš sparčiausiai augančių „Android“ programinės įrangos sričių. Mobilieji žaidimai, tokie kaip „Pokemon GO“, kurie pasiekė pasaulinį populiarumą, naudojasi „Android“ platformos geografiniais ir judesio jutikliais, kad suteiktų vartotojams išplėstinę žaidimo patirtį. „Pokémon GO“ tapo puikiu pavyzdžiu, kaip technologijos, tokios kaip „Android“ ir papildyta realybė, gali sujungti fizinį pasaulį su virtualiais elementais, taip suteikiant vartotojams naują ir įtraukiančią žaidimo patirtį [17].

* Mobilieji mokėjimai ir e-komercija

„Android“ platforma taip pat tapo pagrindine mobiliųjų mokėjimų ir e-komercijos platforma. Programos, tokios kaip „Google Pay“, „PayPal“ ir „Amazon“, suteikia galimybę vartotojams atlikti saugius ir greitus finansinius sandorius naudojant mobiliuosius įrenginius. „Google Pay“, pavyzdžiui, leidžia vartotojams saugiai atlikti pirkimus ir atlikti mokėjimus, nesinaudojant kreditinėmis kortelėmis, tiesiogiai iš jų „Android“ telefonų. Ši funkcija tapo labai svarbi šiuolaikinėje e-komercijoje ir kasdieniniuose pirkimuose, taip pat prisideda prie pasaulinio mobiliųjų mokėjimų populiarumo.

* Švietimo technologijos ir mokymasis

Mokymosi programos, tokios kaip „Duolingo“, tapo itin populiarios, nes jos naudoja „Android“ technologijas kalbų mokymui. „Android“ suteikia kūrėjams galimybę kurti interaktyvias mokymosi programas, kurios pritraukia vartotojus naudodami žaidimus, užduotis ir balus. Puikus to pavyzdys – ta pati „Duolingo“ programėlė, leidžianti vartotojams mokytis naujų kalbų smagiu ir efektyviu būdu, kuri tapo viena populiariausių švietimo programų visame pasaulyje [15].

1. Programinių kalbų apžvalga „Android” programų kūrimui

Paveikslėlis, kuriame yra ekrano kopija, tekstas, diagrama, dizainas

Automatiškai sugeneruotas aprašymas

Programavimo kalbos, kurios iki pat šios dienos buvo naudojamos dažnu atveju yra gan senos kalbos, kurios yra jau užsitvirtinusios savo vietą rinkoje, bet tai nereiškia, kad jos yra blogos, priešingai - dažnu atveju jos yra stabilios kalbos su dideliu funkcionalumu. Naujos kalbos, kurios bando įsitvirtinti kaip pagrindinė šios operacinės sistemos programėlių kūrimo kalba bando ištaisyti senbuvių problemas ir taip užimti „Android” programų kūrimo pagrindinės kalbos poziciją.

Populiarumą išlaikanti yra „Java” kalba, kuri sukurta 1995 metais. Kalbos tikslas buvo tai, kad viena karta sukūrus programą ją būtų galima naudoti bet kokioje platformoje. Tai buvo įgyvendinta panaudojant „JVM” tai yra „Java virtual machine”[[8]](#footnote-8). Kalba yra objektinio programavimo palaikytoja, tuo metu visi rinkosi objektinį programavimą kaip pagrindinę architektūrą programų kūrimui. „Google” kurdama pasirinko „Java” programavimo kalbą dėl keleto priežasčių:[25]

* Rinkoje jau buvo daug programuotojų suprantančių šią kalbą, todėl buvo galima pasirinkti tinkamus specialistus.
* Nemažai mobilių įrenginių naudojo „Java ME” tai yra platforma skirta kurti mobilų kodą. Programos naudojančios šią platformą lengvai valdė sensorius, valdiklius bei kitus mobiliuosius įrenginius.
* Daug įvairių įrankių ir bibliotekų, kurias galima panaudoti programėlių kūrimui.
* Vienas iš esminių privalumų yra „Java” virtuali mašina, kuri leidžia panaudoti tą pati kodą daugeliui skirtingų įrenginių. Virtuali mašina paprastą kodą kompiliuoja į tai sistemai priimtina kodą. Taip pat tai veikia kaip apsauga, kad neteisingai veikiančios komandos neparalyžiuotų sistemos darbo.
* Negalima rankiniu būdu valdyti nuorodų į atmintį, dėl ko programuotojai yra apsaugoti nuo netvarkingos atminties ir jos suvartojimo.

Kotlin programavimo kalba, kuri šiuo metu oficialiai pakeičia „Java” atsirado ne taip ir senai - 2010 metais. Šios programavimo kalbos sukūrimas grindžiamas tuo, kad reikėjo kalbos, kuri galėtu daryti tai ką daro „Java” , bet būtų tvarkingesnė, paprastesnė interpretuoti programuotojui ir tuo pačiu išspręstu „Java” kalbos problemas.[22][23]

* Tai yra pakankama nauja kalba, kuri neturi tiek daug bibliotekų ir įrankių, dėl to kartais gali būti sunku rasti sprendimą iškilusiai problemai.
* Tai moderni kalba, su daug modernių savybių tokių kaip operatorių užklojimas ar paprasta kelių procesorių panaudojimo galimybė.
* „NullPointer” klaidos yra išvengiamos , kadangi kompiliatorius automatiškai šias problemas išsprendžia
* Ši programavimo kalba taip pat atsižvelgia į tai, kad daug sistemų bus perimta iš „Java” dėl to galima nesudėtingai panaudoti su senesniu kodu ir programa nesukels problemų naudotojui.
* Našumo atžvilgiu ši kalba yra lėtesnė nei „Java”, bet tai nėra žymus spartos praradimas.
* Taip pat naudoja „JVM” kodo kompiliavimui ir bazine kalba.

Kitos kalbos, kurios taip pat naudojamos programuojant „Android” sistemos programėles yra „Rust”, „C++”, „C# ir „Xamarin”.Dažnai šios kalbos naudojamos tam, kad sukurti programėles, kurios būtų nepriklausomos nuo platformos, arba tam, kad būtų galima valdyti sensorius ar kitą įranga, kurią galima valdyti tik žemo lygmens programavimo kalbomis. Dažniausiai to nereikia ir pakanka tiesiog pasinaudoti jau sukurtais sistemos įrankiai, kad būtų galima gauti norimą funkcionalumą. Nors kalbant apie „C++“ kalbą kuri gali valdyti žemo lygmens prietaisus, ir palaiko tarp platformų kūrimą, naudojant bibliotekas ir karkasus. „C#“ taip pat elgiasi kaip ir negali valdyti atminties ar žemo lygio prietaisų „C++”, tik priedo yra naudojama „Xamarin” karkasas.[20][24]

1. Programėlės kūrimas

Kaip anksčiau minėta, „Android” yra tikrai populiari ir išlaikanti savo populiarumą operacinė sistema. Dėl to programėlių kūrimas šiaip operacinei sistemai turi didelę įmonių ir bendruomenės paramą, taip pat ir daug sukurtų sprendimų ar reikalingos pagalbos norint sukurti savo pirmąją programėlę. Knygoje apie „Android“ programėlių kūrimą yra išsamiai aprašomas šiai sistemai pritaikytos programėlės kūrimas. Norint pradėti kurti savo pirmąją programėlė reikia parsisiųsti „Android Studio”. Tai oficiali kūrimo platforma šios operacinės sistemos programėlėms. Ją galima rasti: https:[//developer.android.com/studio.](https://developer.android.com/studio)

Paveikslėlis, kuriame yra tekstas, ekrano kopija, programinė įranga, Multimedijos programinė įranga

Automatiškai sugeneruotas aprašymas

pav. Android Studio programėlės kūrimo aplinka

Susiinstaliavus programinę įranga galima pradėti kurti pirmąją programą, bet prieš tai reikia nuspręsti ar norime testuoti sukurtą programą asmeniniame telefone arba yra galimybė naudotis emuliatoriumi, kuris yra įrašomas kartu su programine įranga. Rekomenduoja naudoti emuliatorių[[9]](#footnote-9), kadangi yra galima ištestuoti skirtingas operacinės sistemos versijas ir daug skirtingų įrengimų. Bet būtiną įspėti, kad emuliatorius yra linkęs užstrigti, tad rekomenduojama kompiuterio kurį naudosi neapkrauti nereikalingomis programomis. Taip pat reikia nuspręsti kokia kalba norime programuoti - Java ar Kotlin.

Renkantis kalba reiktu įsivertinti keletą dalykų:

* Ar kuriama programa yra skirta senoms operacinės sistemos versijoms?
* Ar yra planuojama palaikyti seną kodą?
* Ar užimama vieta ir našumas yra svarbu?
* Ar kūrimo laikas ir kodo pakartotinis panaudojimas yra prioritetas?
* Ar kodo vientisumas, paprastumas saugumas yra prioritetas?

Priklausmai nuo to kaip atsakome į šiuos klausimus galime rinktis kalbą. Jei reikia didesnio našumo, ar labai senų sistemų palaikymo tada geriau rinktis Java, bet jei reikia saugumo, neplanuojama palaikyti labai senų sistemų ir vietos turim nemažai, kūrimo laikas ir kodo panaudojamumas yra svarbu, tada vertėtų rinktis modernesnę programavimo kalbą Kotlin.[18][21]

Išsprendus pagrindiniu du klausimu galima pereiti prie programavimo ir programėlės dizaino. „Android Studio” naudoja XML programavimo kalba ar „Jetpack Compose”. XML yra senesnė kalba skirta aprašyti ir sudėlioti vartotojo sąsają, kai tuo tarpu „Jetpack Compose” yra daug naujesnė ir veikia principu „Material design”[[10]](#footnote-10). Sudėtingumo atžvilgiu XML yra šiek tiek lengviau išmokstama, bet jos kodas gali greitai išsipūsti, norint sukurti „Responsive”[[11]](#footnote-11) dizainą. Tuo tarpu „Jetpack Compose” yra deklaratyvi kalba, kuri yra šiek tiek sudėtingesnė, bet kadangi tai nauja kalba, jai trūksta palaikymo, bet norint aprašyti vartotojo sąsajos elementus reikia mažiau kodo ir tai yra deklaratyvus kodas, kas reiškia galima jo rasti kartu su reguliariu kodu ir tai kas parašomą taip ir veikia sąsajos objektas, nereikia nurodyti, kaip pasiekti norimą veiksmą, jis tiesiog veiks pasirinktas objektas. Taip pat kuriant programėlės dizainą yra galimybė, jį kurti net nežinant XML ar „Jetpack Compose” nes programavimo aplinka palaiko nuvilkimo ir padėjimo funkcija vartotojo sąsajos kūrimui. [24]

Kitas esminis dalykas apie programavimą su „Android Studio” ir pasirinkta programavimo kalba yra kompiliavimo programos. Tam, kad nereiktu rašyti ilgų komandų kompiliatoriui yra naudojamos viena iš dviejų projekto kūrimo programų - „Gradle” arba „Maven”. Labai didelių skirtumu tarp šių programų nėra, bet kaip pradedančiąjam naudotojui galima naudoti „Maven” dėl to, kad ji paprastesnė sukonfigūruoti. Vertinant projekto apimtį jei žinoma, kad projektas bus didesnės apimties ir reikalaus daugiau resursų rekomenduojama naudoti „Gradle”, dėl savo spartos kompiliuojant, nes ši programa iš naujo nekompiliuoja senų failų, tik naujus, kai tuo tarpu „Maven” kompiliuoja viska. Taip pat „Gradle” yra lankstesnė ir turi daugiau funkcionalumo, kas padaro ją patrauklia didesniems projektams.[26]

1. Įdomūs faktai, susiję su programų kūrimu „Android“

Per ilgesnį programėlių kūrimo šiai operacinei sistemai laikotarpį ir šios operacinės sistemos gyvavimą atsirado daug įdomių faktų ir tai keletas iš jų [8][9]:

* „Android” būdama pakankamai sena sistema turi daug operacinės sistemų variacijų, bet daug programėlių gali veikti ir su senom versijom.
* Ankstesnė problema su šios operacinės sistemos atnaujinimų trūkumu ir spartos problemomis buvo dėl prastos optimizacijos, bet dar didesnę problemą kėlė procesorių kūrėjai, kurie pakankamai greitai nustodavo palaikyti procesorių ir pereidavo prie naujų kūrimo, dėl to sistemos kūrėjai nebeatnaujindavo savo operacinės sistemos ir taip likdavo daug neišspręstų problemų.[19]
* Android kompanija 2005 metais pardavė savo sukurtą sistema „Google” kompanijai už 50 mln. Dolerių sumą.
* Prieš parduodant savo sistemą „Google”, ši operacinė sistema buvo pasiūlyta „Samsung” įmonei, bet jiems „Android” pasirodė neįdomus projektas.
* Pradžioje „Android” ;buvo kuriamas skaitmeninėms kameroms, bet vėliau tai buvo pakeista į mobiliuosius įrenginius.
* „Android” buvo kuriamas naudojant daug kalbų, tokių kaip „C”, „C++”, „Java”, ir net „Assembler” bei daugeliu kitų.
* „HTC Dream” buvo pirmasis telefonas, kuris naudojo „Android” operacinę sistemą.
* Beveik 90% praleisto prie ekranų laiko yra praleidžiama programėlėse.
* Vidutinis naudotojas turi daugiau nei 80 programėlių, nors daugiausiai panaudoja 9 per diena ir 30 per mėnesį, kol kitos programėlės telefone yra tiesiog nenaudojamos.
* Mobilieji žaidimai labai populiarūs, net 40% telefonų naudotojų juos žaidžia, o 80% iš jų yra „Android” naudotojai.[19]
* „Android” naudotojai turi net 1.5 karto daugiau programėlių pasirinkimo nei „IOs” naudotojai.
* Per mėnesį yra sukuriama net 100,000 naujų programėlių.
* Net 50% naujų pirkimų yra atliekama telefonu naudojant kokią nors programėlę.
* Net 80% programėlių, kurios bus įrašytos mobiliajame įrenginyje bus nebenaudojamos po 70val.

1. Išvados

„Android” ir šios operacinės sistemos palaikomos, bei vartotojų naudojamos programėlės ir toliau išlieka didžiausi rinkos žaidėjai užimdami net 70% viso pasaulio rinkos ir išlaikančios savo dominavimą. Artimiausiu metu nenusimato, kad „Google” ir jos valdoma „Android” operacinė sistema bus iš rinkos išstumta net ir didžiausias konkurentas „Apple”, kuris valdo tik 30% rinkos artimiausiu metu neturi galimybių pasiekti „Android” populiarumo. Toliau pateikiamos šio darbo išvados:

* Dėl gero prieinamumo, užimamo plataus kainų segmento programėlių kūrimas ir „Android” operacinės sistemos populiarumas ir toliau išliks didelis.
* Dėl savo gebėjimo naudoti „JVM” „Android” gali būti pritaikomas veikti plačioje mobiliųjų įrenginių aibėje.
* Norint kurti programėles yra didelis platformos palaikymas ir plati dokumentacija, kuri leidžia pradedančiajam nesunkiai mokytis kurti savo programėles.
* Nors „Java” buvo naudojama ilgai ir turi užtikrintą vietą rinkoje, su daug bibliotekų ir karkasų, bet poreikis ištaisyti jos užsilikusias klaidas ir pereiti prie modernesnių kalbų suteikė šansą atsirasti bei išpopuliarėti „Kotlin”.
* „Kotlin kalba gali palaikyti seną „Java” kodą, kas padeda nesunkiai pereiti prie modernesnės kalbos.
* Daug programėlių, kurios yra sukurtos ir patalpintos „Google Play” platformoje yra nemokamos.
* Daug programėlių panaudojus trumpą laiką vartotojų yra paliekamos likimo valiai ir daugiau nebenaudojamos.
* Buvusios problemos kaip operacinės sistemos strigimai ar programėlės, kurios galėdavo sustabdyti jos veikimą buvo netvarkingai suprogramuotos ar optimizuotos. Bet didelė dalis to kaltės buvo dėl procesorių gamintojų.

1. Literatūra
2. Hayou Wang, Zhe Liu, Jingyue Liang, Narseo Vallina-Rodriguez. Beyond Google Play: A Large Scale Comparative Study of Chinese Android App Market. , 2018.
3. Lionel Sujay Vailshery. Most used Programing Languages among Developers Worldwide as of 2024. Interaktyvus [2024-9-18]. Prieiga per: [https://www.statista.com/statistics/793628/worldwide-developer-survey-most-used-languages/.](https://www.statista.com/statistics/793628/worldwide-developer-survey-most-used-languages/) [žiūrėta 2024-12-14].
4. Ivan Trogrlic. A Step-by-Step Guide to Android App Development. Interaktyvus [2023-9-4]. Prieiga per: [https://decode.agency/article/android-app-development/.](https://decode.agency/article/android-app-development/) [žiūrėta 2024-12-14].
5. Michael Burton. Android App Development For Dummies. , 2017.
6. Arvydas Putna. C# ir Java programavimo kalbų lyginamoji analizė. Interaktyvus, 2015. Prieiga per: <https://portalcris.vdu.lt/server/api/core/bitstreams/68d0caa6-7173-4d9b-acfb-043ac105d869/content.> [žiūrėta 2024-12-13].
7. Subham Bose, Aditi Kundu, Madhuleena Mukherjee, Madhurima Banerjee. A comparative study: Java vs Kotlin programing in Android application development. International Journal Of Advanced Research In Computer Science, vol. 9 (2018). Interaktyvus. Prieiga per: <https://www.semanticscholar.org/reader/b047262d0a95f9d910e5336703c08d1a78902e31.> [žiūrėta 2024-12-13].
8. Moisiejus Lukas. Programėlių išmaniesiems įrenginiams kūrimo technologijų analizė. Interaktyvus, 2017. Prieiga per: [https://epubl.ktu.edu/object/elaba:22833611/.](https://epubl.ktu.edu/object/elaba:22833611/) [žiūrėta 2024-12-13].
9. Android Developers. . Prieiga per: [https://developer.android.com/.](https://developer.android.com/)
10. Kotlin for Android Developers. Interaktyvus. Prieiga per: [https://kotlinlang.org/.](https://kotlinlang.org/)
11. Hanna Esmaeel,. Apply Android Studio (SDK) tools (2015). Interaktyvus. Prieiga per: <https://www.researchgate.net/publication/331673953_Apply_Android_Studio_SDK_Tools>..
12. SEJAL THAKKAR, CHIRAG PATEL, Ved Suthar. PLANT DISEASE IDEDNTIFICATION USING MACHINE LEARNING AND IMAGE  
    PROCESSING. ICTACT JOURNAL ON SOFT COMPUTING, vol. 13 (2023), nr. 4. Interaktyvus. Prieiga per: <https://ictactjournals.in/paper/IJSC_Vol_13_Iss_4_Paper_5_3043_3047.pdf?fbclid=IwY2xjawHKzgtleHRuA2FlbQIxMAABHeAxb5X3TgwTdPHQQmovl8KegaHRf1ebfcC1_Uk1Ad6FP6OJOsTdUQ30hQ_aem_2ekI6liBQr8Tu2Nmk2U3iA.>
13. Dukka Karun Kumar; REDDY, H.S.; BEHERA, Janmenjoy ir kt. Exact greedy algorithm based split finding approach for intrusion detection in fog-enabled IoT environment. Journal of Information Security and Applications, vol. 60 (2021). Interaktyvus. Prieiga per: <https://www.sciencedirect.com/science/article/abs/pii/S2214212621000971?fbclid=IwY2xjawHKzvlleHRuA2FlbQIxMAABHUf3Shbik47_e4A-I3AbnPCb0T2RZXMVa20cpVTHiednVhkEJbW-Zl5n9Q_aem_kq-JZVpuEgvbVWl4lzGVJA.>
14. Dowling, L., Jakeman, P., Norton, C. et al. Adults with Crohn’s disease exhibit elevated gynoid fat and reduced android fat irrespective of disease relapse or remission. Sci Rep 11, 19258 (2021). Interaktyvus. Prieiga per: <https://www.nature.com/articles/s41598-021-98798-9?fbclid=IwY2xjawHK0EpleHRuA2FlbQIxMAABHYXRP1Pfc3G3Q85VoefrjP8tKhbmAk9kYWGagrqY-FpwPgjr_0NewWj1jg_aem_VTcneuAYYpTGtCWqn9FCcQ>
15. Kiranmayi, D., Sharma, A., Prasad, K.P. et al. Development of an Android-Based Application System for Fish Farmers. Agric Res 11, 240–248 (2022). Interaktyvus. Prieiga per: <https://link.springer.com/article/10.1007/s40003-021-00558-8?fbclid=IwY2xjawHK0GJleHRuA2FlbQIxMAABHdJoI_7uaggp8UjFYtWg7YvIpkJZIbYGT3j5v2yGojdha4GXoG-VFKMC6w_aem_gIr970mDpj11X9sHHNygig#Bib1>
16. Martínez-Pérez, B., de la Torre-Díez, I., & López-Coronado, M. (2013). Mobile health applications for the most prevalent conditions by the World Health Organization: review and analysis. Journal of medical Internet research, 15(6), Interaktyvus. Prieiga per: <https://pmc.ncbi.nlm.nih.gov/articles/PMC3713954/?fbclid=IwY2xjawHK0PpleHRuA2FlbQIxMAABHeAxb5X3TgwTdPHQQmovl8KegaHRf1ebfcC1_Uk1Ad6FP6OJOsTdUQ30hQ_aem_2ekI6liBQr8Tu2Nmk2U3iA>
17. Michel M, Pawlaszczyk D, Zimmermann R. AutoPoD-Mobile—Semi-Automated Data Population Using Case-like Scenarios for Training and Validation in Mobile Forensics. Forensic Sciences. 2022; Interaktyvus. Prieiga per: <https://www.mdpi.com/2673-6756/2/2/23?fbclid=IwY2xjawHK0ZBleHRuA2FlbQIxMAABHS_iDzDF_vUDH2nHUzVlRD8DUsq0_EiTkGoREKcSK2dphAi8Nk7y6Yz1nA_aem_W_flJhBUFO3um6gDPNyL6g>
18. Parekh, P., Patel, S., Patel, N. et al. Systematic review and meta-analysis of augmented reality in medicine, retail, and games. Vis. Comput. Ind. Biomed. Art 3, 21 (2020). Interaktyvus. Prieiga per: <https://link.springer.com/article/10.1186/s42492-020-00057-7?fbclid=IwY2xjawHK0fJleHRuA2FlbQIxMAABHYXRP1Pfc3G3Q85VoefrjP8tKhbmAk9kYWGagrqY-FpwPgjr_0NewWj1jg_aem_VTcneuAYYpTGtCWqn9FCcQ#citeas>
19. JetBrains. Comparison to Java. . Prieiga per: <https://kotlinlang.org/docs/comparison-to-java.html#what-kotlin-has-that-java-does-not.> [žiūrėta 2024-12-13].
20. 13 Facts about Mobile App Development. . Prieiga per: [https://www.activate.co.nz/Blog/13-facts-about-mobile-app-development-2022/.](https://www.activate.co.nz/Blog/13-facts-about-mobile-app-development-2022/) [žiūrėta 2024-12-13].
21. Medium. Why Jetpack Compose is Not as Easy as it Looks? (For Real), pp. 1Prieiga per: <https://medium.com/@rahulgohil08/why-jetpack-compose-is-not-as-easy-as-it-looks-for-real-e57039a31179.> [žiūrėta 2024-12-13].
22. B. P. D. Putranto; R. Saptoto; O. C. Jakaria ir W. Andriyani. A Comparative Study of Java and Kotlin for Android Mobile Application Development. In: 2020 3rd International Seminar on Research of Information Technology and Intelligent Systems (ISRITI), pp. 383–388.2020. ISBN null.
23. SCHWERMER, Patrik. Performance Evaluation of Kotlin and Java on Android RuntimePrieiga per: <https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1231573&dswid=-1372.> [žiūrėta 2024-12-13].
24. ARDITO, Luca; COPPOLA, Riccardo; MALNATI, Giovanni ir TORCHIANO, Marco. Effectiveness of Kotlin vs. Java in android app development tasks. Information and Software Technology, vol. 127 (2020), pp. 106374. ISSN 0950-5849. Prieiga per: <https://www.sciencedirect.com/science/article/pii/S0950584920301439.> [žiūrėta 2024-12-13].
25. Jetpack Compose. . Prieiga per: <https://developer.android.com/codelabs/jetpack-compose-basics#6.> [žiūrėta 2024-12-13].
26. Why Google Chose Java for Android. . Prieiga per: <https://medium.com/@kodeguy/why-did-google-choose-java-as-default-application-programming-language-for-android-310319a8ba45.> [žiūrėta 2024-12-13].
27. Gradle Vs Maven. . Prieiga per: <https://www.dhiwise.com/post/kotlin-maven-vs-gradle-which-build-tool-is-right.> [žiūrėta 2024-12-13].
28. Wiki.Geek. Write Once Debug Everywhere. . Prieiga per: <https://wiki.secretgeek.net/write-once-debug-everywhere.> [žiūrėta 2024-12-13].

1. IDE – integruota plėtros aplinka [↑](#footnote-ref-1)
2. „Android“ programos paketas [↑](#footnote-ref-2)
3. angl. Extensible Markup Language [↑](#footnote-ref-3)
4. Virtuali realybė [↑](#footnote-ref-4)
5. Dirbtinis intelektas [↑](#footnote-ref-5)
6. Liet. Parašykime viską vienu kartu ir gerai [↑](#footnote-ref-6)
7. Liet. Parašykime viską vienu kartu, tačiau visur taisykime klaidas [↑](#footnote-ref-7)
8. Java virtuali mašina [↑](#footnote-ref-8)
9. 9 Virtualus įrenginys, skirtas testuoti Android sukurtas programas [↑](#footnote-ref-9)
10. Dizaino kalba [↑](#footnote-ref-10)
11. Prisitaikantis [↑](#footnote-ref-11)