# **MIPT egzamino užduotis**

1. **Apibrėžkite programinės įrangos testavimo procesą. Kokia jo paskirtis?**

Programinės įrangos testavimo procesą sudaro tokie testai kaip „Functional testing“, „User interface testing“, „Compatibility testing“, „Interface testing“, „Network testing“, „Performance testing“, „Security testing“, „Field testing“ ir „Interrupt testing“. Jie yra skirti užtikrinti, kad programinė įranga visais atvejais veikia sklandžiai, veikia numatytose aplinkose ir atitinka visus užsakovo reikalavimus.

1. **Išvardinkite ir apibūdinkite keletą (3-5) testavimo tipų?**

Functional testing – testavimas patvirtinant reikalavimus, kuriuos programa turi atlikti, pvz.: „Wolt“ programoje pasirinkus Kauną turi rodyti Kaune esančius restoranus.

User interface testing – tikrina ar vartotojo sąsaja veikia teisingai, t. y. ar matosi visas tekstas ir mygtukai, ar interaktyvus elementai veikia atitinkamai, kaip programa atrodo skirtinguose ekranuose ir telefonuose.

Compatibility testing – kaip ir „User interface testing“, tikrina suderinamumą su skirtingų dydžių ekranais, raiškomis, bet taip pat tikrina ar programa veikia skirtingose „Android“ OS versijose/telefonuose ir ar programa sėkmingai prisijungia prie interneto

Interface testing – tikrina sąveiką su kitomis programomis („Google Maps“, „Facebook“, t.t.), mikrofono naudojimąsį tekstui įvesti, kameros naudojimą „QR code“ nuskaityti.

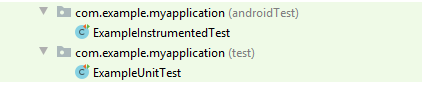
1. **Paaiškinkite pagrindinį TDD (*Test Driven Development*) principą. Kaip manote, kodėl šis metodas ne visada pasiteisina?**

Kiekvienai programos funkcijai sukuriami „test cases“, kurie tada yra išbandyti. Jei testai neišlaikomi, kodas perrašomas siekiant išlaikyti testą (TL;DR, tests before code). TDD neturėtų būti taikomas vartotojo sąsajai kurti, kadangi sukurti ir palaikyti „GUI“ testus yra brangu ir neefektyvu, o „GUI“ dažnai keičiasi. Taipogi, testai, kaip ir programos kodas, turi būti palaikomi ir programa pasidaro sudetingesnė, nei turėtų būti.

1. **Kokio tipo testai yra patalpinami šiose Android aplikacijos projekto direktorijose:**

„Instrumented Test“ testuoja sukurtą programą emuliatoriuje ar įrenginyje. Jie yra naudingi tada, kai norima išbandyti programą kuri veikia kitaip skirtinguose kontekstuose. Naudojami norint testuoti vartotojo sąsają.

„Unit Test“ testuoja sukurtą programą nenaudojant emuliatorių ar įrenginių. Jie yra naudojami norint patikrinti programos funkcionalumą.



1. **Kokiomis proporcijomis (testų skaičiumi) rekomenduojama turėti *Unit*, *Integration* bei *UI* testus. Kodėl?**

Reikomenduojama testuoti 70:20:10 (Unit:Integration:UI) proporcijomis, kadangi sudėtingesnius testus sunkiau palaikyti, pataisyti ir jie trunka žymiai ilgiau nei Unit testai.