

Problēmas nostādne

Programmatūras testēšanas mērķis ir identificēt sistēmā esošās kļūdas un nodrošināt to, ka sistēma ir uzticama. Pieejamās testēšanas tehnikas nespēj nepieciešamo uzticamību nodrošināt pilnīgi, jo vēl joprojām, neskatoties uz ieguldītajiem resursiem programmatūras testēšanā, sistēmā atklājas vēl nepamanītas kļūdas.

Programmatūras testēšana

Programmatūras testēšana ir sistēmas vai tās komponenta novērtēšana ar nolūku noskaidrot, vai sistēma vai tās komponents atbilst noteiktām prasībām. Testēšanas laikā sistēma tiek darbināta, lai identificētu jebkādas nepilnības, kļūdas vai iztrūkstošās prasības, balstoties uz esošām, iepriekš noteiktām vēlmēm.

Specifikācijas vadīta testēšana

Specifikācijas vadīta testēšanas tehniku dēvē arī par melnās kastes testēšanas tehniku, jo šajā gadījumā testētājam nav zināšanu par sistēmas vai komponenta struktūru. Testēšana, testa piemēru, ievaddatu veidošana un sistēmas izvērtēšana atbilstībai tiek balstīta uz pieejamo specifikāciju.

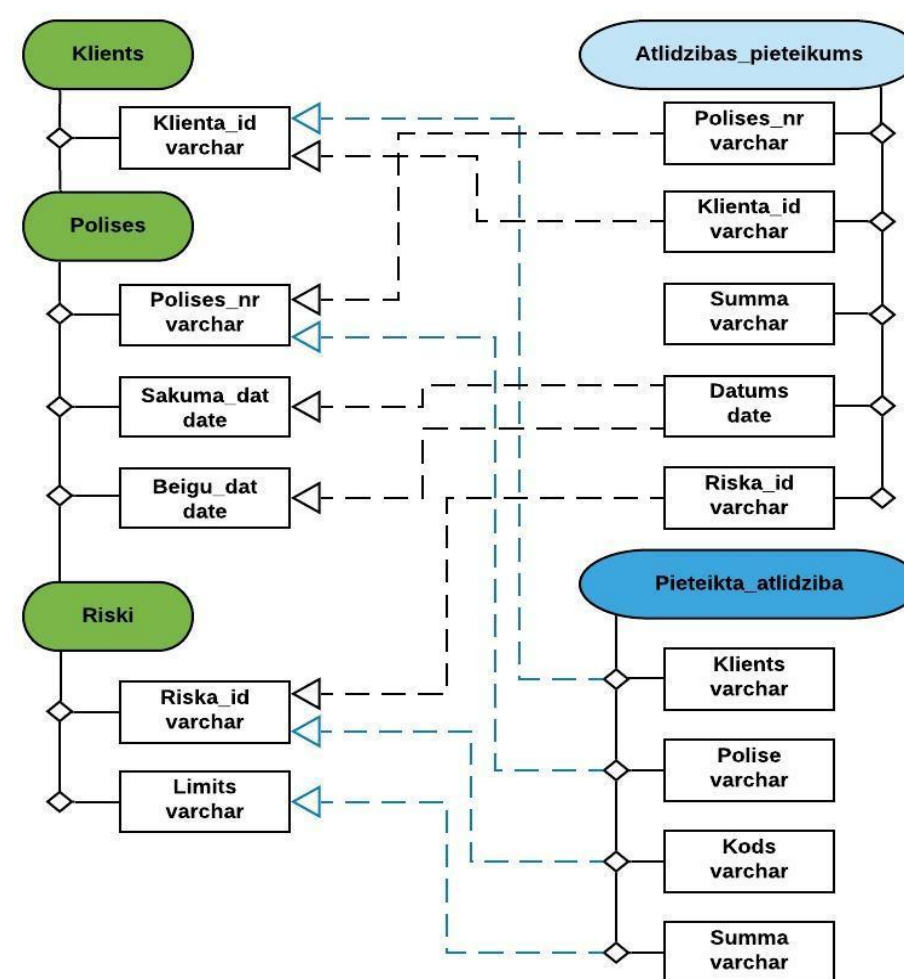
Darba mērķis

Piedāvāt un praktiski pārbaudīt jaunu testēšanas tehniku, kas veicinātu uzticamas programmatūras nodrošināšanas mērķa sasniegšanu.

Risinājuma ideja

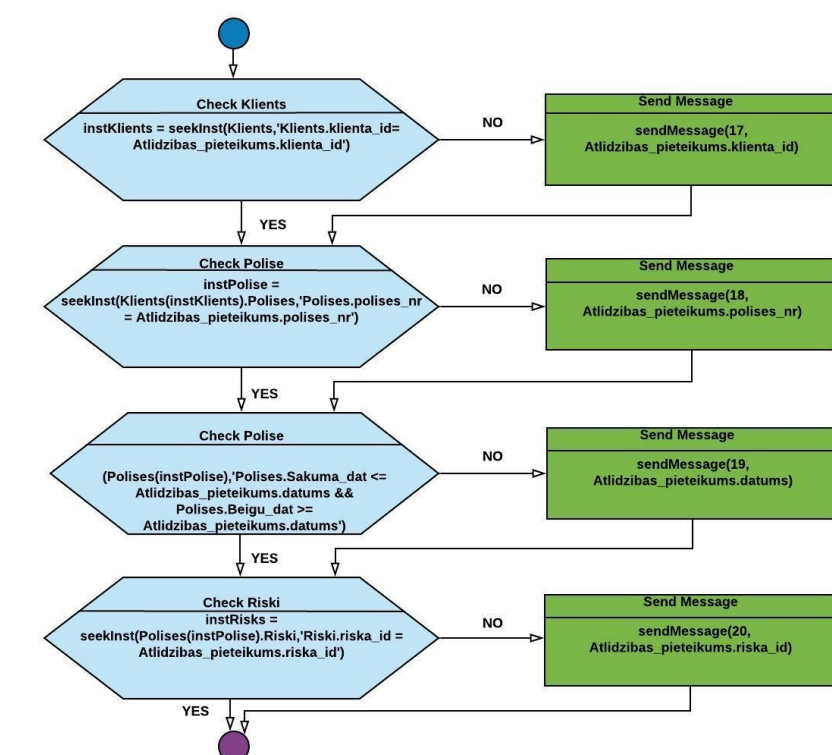
Vispirms izstrādā sistēmas specifikāciju ar augstu abstrakcijas līmeni, kur specifikācija sastāv no datu objektiem un nosacījumu jēdzieniem. Datu objekti apraksta reālās pasaules objektus un nosacījumi ietver prasības, kas datu objektu atribūtu vērtībām jāapmierina, lai datu objekts tiktu atzīts par korektu. Tiek aplūkoti sistēmā ievadāmie datu objekti un ar datu objektu nosacījumiem pārbaudīts, vai tie ir korekti. Ja dati ir korekti, tie tiek

saglabāti datu bāzē, bet, ja dati ir nav korekti, datu īpašnieks saņem paziņojumu un var veikt datu koreģēšanu, un atkārtoti ievadi sistēmā. Datu objektu kontrole tiek veikta divos līmeņos. Sintaktiskā kontrole pārbauda, vai ievadāmo datu objektu atribūtu vērtības atbilst atribūtu vērtību sintaksei. Semantiskā kontrole pārbauda, vai datu objektu atribūtu vērtības atbilst citu datu objektu vērtībām. Risinājumā piedāvāts salīdzināt ievadāmo datu objektu un attiecīgi datu bāzē saglabāto datu objektu atbilstību.



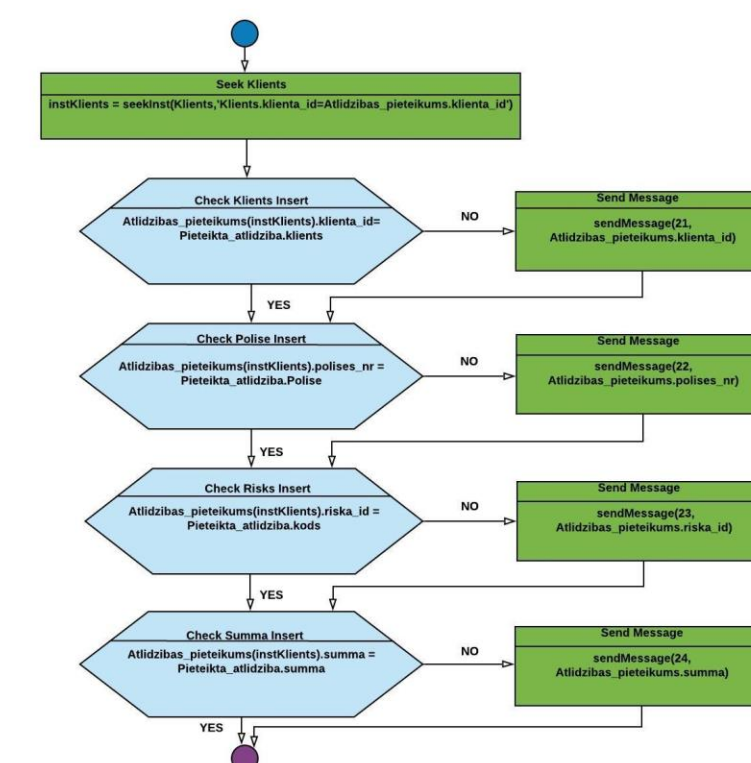
1. att. Datu objekta piemērs

Ievadāmo datu objektu kontroli var panākt, izmantojot loģiskās izteiksmes. Loģisko izteiksmju operandi attiecīgi šeit ir datu objektu atribūtu vērtības, bet operācijas var tikt veidotas jebkurā programmēšanas valodā.



2. att. Datu objekta kvalitātes novērtēšanas priekšnosacījumi

Katrs datu objekta klases objekts tiek pārbaudīts ar izveidoto nosacījumu palīdzību. Pareizības kontrole tiek veikta ar priekšnosacījumu palīdzību pirms reālās datu ievades. Tās laikā pārbauda, vai ievadāmo datu objektu atribūtu vērtības atbilst atribūtu vērtību sintaksei un datu objektu atribūtu vērtības atbilst citu datu objektu vērtībām.



3. att. Datu objekta kvalitātes novērtēšanas pēcnosacījumi

Kad dati ievadīti sistēmā, tiek veikta datubāzē ierakstīto datu pārbaude ar pēcnosacījumu palīdzību. Tās laikā pārlicinās, ka ievadītie dati tiek korekti saglabāti.