****

Latvijas Universitātes Vadības un uzņēmējdarbības mācību centrs

(LU VUMC)

**JAVA PROGRAMMĒŠANAS PAMATI**

2021

ESF projekts Nr. 8.4.1.0/16/l/001  
‘’Nodarbināto personu profesionālās kompetences pilnveide”



# Nodarbība 12

## Terminu vārdnīca

Tīrs kods : tāds kods, kuru ir viegli saprast un viegli uzturēt. Kods var būt ģeniāls un ļoti efektīvs, bet, ja tas ir nesaprotams, tas nebūs arī viegli maināms un uzturams. Pastāv daudz nesaprotama un neuzturama koda piemēru; programmētāji pat sacenšas šāda koda radīšanā. Piemēram, ir tāds ikgadējs konkurss IOCCC -- The International Obfuscated C Code Contest. Te būs šāda koda piemērs, 2006.gada labākā īsākā programma, kuras autors ir Oscar Toledo; tā risina zirga gājienu uzdevumu šahā, vai tas spēj apmeklēt visas 64 šaha dēļa rūtiņas bez atkārtošanas. char \*e,t [366 ],\*f ,\*g, \*h,\* i;d, m ; main (c,b) char \*\*b;{ for( ;d[t] =d%3 ?60<d &300 >d&6 <d % 30?0 :32: d%30 ?32: 10 , 366> ++d;) ;for (g=3\* atoi (\*++ b) + 34+t; i=f= "\1" "\7" "(d" "\177" "yX" "\34" ,e=g ; ) for( \*e++ =++m/ 10 + 48,g =c=0 ,\*e = 48+m %10; h=e- 65+\* i,d= \*h,\* i++; f-=8 ,g=d <c?c =d,h: g)for (; d -=!! h[\*f ++-64 ],\*f ;) ; puts (t);}

Technical debt : Tehnisks parāds. Arī zināms kā projektēšanas parāds (design debt) vai koda parāds (code debt). Tā ir koncepcija programmatūras izstrādē, kura apraksta papildu pārstrādes izmaksas, nepieciešamība pēc kuras rodas dēļ tā, ka konkrētā brīdī tiek izvēlēta vieglāka ierobežota risinājuma izstrāde tā vietā lai radītu labāku, pilnīgāku risinājumu, kurš prasītu vairāk laika. Līdzīgi kā ar fiskālu parādu, ja tehnisks parāds netiek atmaksāts, tas var "uzkrāt procentus", tādējādi padarot izmaiņu izstrādi arvien grūtāku. Tāpat, līdzīgi kā fiskāls parāds, tehnisks parāds ne vienmēr jāuztver kā slikta lieta, un dažreiz tā parādīšanas un esamība projektā ir pamatota.

Refactoring : Refaktorēšana. Tas ir esošā koda pārstrādes un pārstrukturēšanas process (faktoringa mainīšana), kurš paredz nemainīgu iekšēju koda loģiku un uzvedību. Refaktorēšanas mērķis ir koda dizaina, struktūras uzlabošana, saglabājot nemainīgu tā funkcionalitāti. Refaktorēšanas iespējamas priekšrocības ir : uzlabota koda lasāmība un saprotamība; samazināta koda sarežģītība; uzlabota koda uzturamība. Līdz ar to galvenais mērķis ir tīrāks kods. Otrs svarīgs mērķis, kura dēļ mēdz uzsākt refaktorēšanas procesu, ir koda veiktspēja : programmatūras inženieri sastopas ar nepārtrauktu izaicinājumu radīt programmas, kas patērē mazāk atmiņas un strādā ātrāk un efektīvāk.

Java klase : Klases un objekti ir pamata koncepti objekt-orientēta programmēšanā, kuru jēdziens un būtība apgrozās ap un pietuvojas reālas dzīves objektiem. Java klase, tāpat kā jebkuras OOP valodas klase, ir programmētāja definēts prototips, no kuras tiek veidoti objekti. Tas definē un atspoguļo atribūtu un metožu kopu, kas piemīt visiem šāda tipa objektiem.

Metode : tas ir koda bloks ar nosaukumu, kurš tiek izpildīts tikai tad kad to izsauc. Tas var saņemt datus izsaukuma brīdī caur parametriem, ja vien tādi ir definēti. Metode var atgriezt rezultātu, vai arī neatgriezt, atkarībā no metodes definīcijas. Metodes objektā definē tā uzvedību, kā arī var piekļūt un operēt ar objekta atribūtiem (laukiem). Metožu mērķis ir funkcionalitātes atkārtota izmantošana.