📖 java-6-nodarbiba-cikli-1-uzdevums-obligats.md

Tēma 6

Uzdevums 1 - Izveidot reizināšanas tabulu.

Priekšnosacījumi

- pirms sākt darbu pārliecinamies ka versiju sistēmas "Work area" nav mainītu failu. Iespējams izpildam commit vai rollback tiem.
- Izpildām "Project update" izmantojot versiju sistēmas tool. Karstais taustiņš CTL + K.

Kopsavilkums

- izveidot reizināšanas tabulu kuru izvada teksta veidā:

```
   1   2   3
---------------
   1   2   3
   2   4   6
   3   6   9
```

Soļi:

- Izveidojam jaunu klasi MultiplicationTable
- izveidojam metodes:

    - calculate().

      metode aprēķina 2 skaitļu reizinājumu

```
int calculate(int x,int y)
```

- printHeaders().

  metode izdrukā virstaksta rindu

```
void printHeaders(size)
```

- printDetails()

  metode atgriež teksta vērtību reeizinājumiem ar 'row' t.i. horizontālo rindu ar reeizinājumiem

```
String printDetails(size, row)
```

- ievietojam main metodi un tajā izsaucam printHeaders() un printDetails() metodes

```
int size = 10;
printHeaders(size);
for(int i=0;i<size;i++){
   printDetails(size, i);
}
```

Uzdevums 2 - Vienādojuma risināšana ar brute force metodi

Kopsavilkums

dots šads vienādojums:

(constant) = (one) * x + (two) * y + (three) * z

gribam implementēt metodi bruteForceEquationSolver(one, two, three, constant),
kura risina šo vienādojumu ar Brute Force metodi, tas ir pārmeklējot visas x,y,z vērtības no 1 līdz kādam maksimumam.

piemēram, ja metodei nodod šādus argumentus (2, 3, 4, 42), tiktu meklēts risinājums šādam vienādojumam 2x + 3y + 4z = 42.

metode atgriež šādu rindu

" x: xSolution y: ySolution z: zSolution " vai arī "risinājums nav atrasts"

Soļi

- Izveidojam jaunu pakotni student_xxxxx_xxxxx.homework_6.task1
- Izveidojam jaunu klasi BruteForceEquationSolver iepriekš izveidoyaajā pakotnē.
- Klasē ievietojam konstanti MAX_VALUE kura ierobežo iterāciju maksimālo vērtību.

```
final int MAX_VALUE = 1000; // palielinot šo vērtību programmas izpilde palēninās.
```

- Klasē ievietojam sekojošu metodi.

```
public static String bruteForceEquationSolver(int one, int two, int three, int constant){}
```

šī metode ar Nested Loops metodi meklēe risinājumu šim vienādojumam.

Nested Loops nozīmē trīs iekļautus for ciklus

```
    x = 1 .. MAX_VALUE
       y = 1 ... MAX_VALUE
```

```
            z = 1 .. MAX_VALUE
               TEST_EQUATION
```

vienādojums ir atrisināts, ja

one * x + twoo * y + three*z == constant

- Ievietojam main metodi
- main() metodē
    - izsaucam bruteForceEquationSolver() ar kādām konkrētām koeficientu (argumentu) vērtībām.
    - ievietojam izdruku, kura parāda metodes atgriezto vērtību (vienādojuma atrisinājumu vai arī ka vienādojumam nav risinājuma pie dotā MAX_VALUE).