

Kotitehtävä 5 Deadline 12.10. klo 11:45

Maksimipisteet 10

Valitse tehtäviä niin, että saat niistä maksimissaan 10 pistettä

Tehtävä 1: Arrayn komponenttien tulostus (1p)

Too Many Words (Try It Out p. 303)

In a fit of shameless narcissism, I made a rough count of the number of words in pages 1 to 7 of this book's previous edition.

```
int[] words = {0, 296, 342, 405, 363, 350, 323, 101};
```

Write a program that prompts the user for a page number and responds with the count of words on that particular page.

Olio_ohjelmointi - kt_5/src/kt_5/TooManyWords.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

TooManyWords.java

```
1 package kt_5;
2
3 import java.util.Scanner;
4
5 public class TooManyWords {
6
7     public static void main(String[] args) {
8         // Luodaan taulukko tallentamaan sanojen määrän jokaisella sivulla
9         int[] words = {0, 296, 342, 405, 363, 350, 323, 101};
10
11         // Luodaan Scanner-olio käyttäjän syötteen lukemiseen
12         Scanner scanner = new Scanner(System.in);
13
14         // Kysytään käyttäjältä sivunumero
15         System.out.println("Anna sivunumero: ");
16         int pageNumber = scanner.nextInt();
17
18         // Tarkistetaan, onko sivunumero kelvollinen
19         if (pageNumber < 1 || pageNumber > 7) {
20             System.out.println("Kelvoton sivunumero.");
21             System.exit(1);
22         }
23
24         // Haetaan sanojen määrä pyydetyltä sivulta
25         int wordCount = words[pageNumber - 1];
26
27         // Tulostetaan sanojen määrä pyydetyltä sivulta
28         System.out.println("Sivulla " + pageNumber + " on " + wordCount + " sanaa.");
29     }
30 }
31
```

Console

<terminated> TooManyWords [Java Application] C:\Users\journi\p2\pool\plugins\org.eclipse.justi.openjdk.b...
Anna sivunumero:
6
Sivulla 6 on 350 sanaa.

Tehtävä 2: modify the for loop (1p)

Spell It Backward (Try It Out p. 303)

Modify the code in Listing 11-1 so that it displays the rooms in reverse room number order (starting with Room 9 and working down to Room 0).

Tehtävä 3: komponenttien lisääminen arrayhin ja komponenttien arvojen käyttö toimintoon (2p)

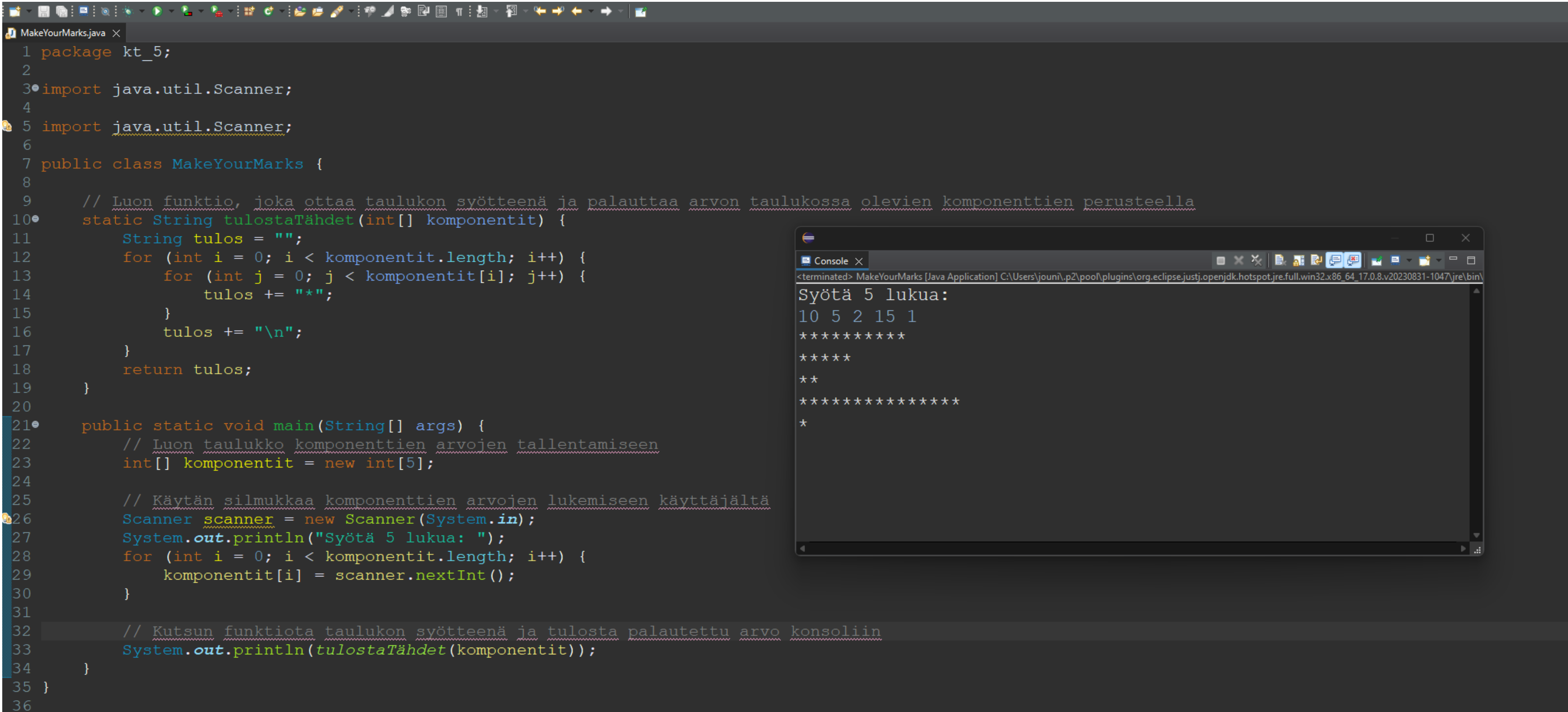
Make Your Marks (Try It Out p. 303-304)

First the user types five numbers on the keyboard. Then the program displays five lines of characters, each line having the same number of asterisks as one of the user’s numbers. A run of the program might look like this:

```
>> 5
>> 3
>> 9
>> 0
>> 10
```

```
*****
***
*****

*****
```



Tehtävä 4: Selitä kuvan avulla (2p)

Piirrä kuva

Katso listausta 11-3. Piirrä kuva siitä, miten for-luuppi toimii. Voit myös yrittää selittää loopin omin sanoin. Mitä tapahtuu missäkin luupin vaiheessa? Selitä myös miten enhanced for-loop (parannettu silmukka tai for each toistolause) eroaa tavanomaisesta for loopista.

LISTING 11-3:	Get a Load o’ That for Loop!
<pre>package com.example.hotel; import static java.lang.System.out; public class EnhancedShowGuests { public static void main(String[] args) { int[] guests = {1, 4, 2, 0, 2, 1, 4, 3, 0, 2}; int roomNum = 0; out.println("Room\tGuests"); for (int numGuests : guests) { out.print(roomNum++); out.print("\t"); out.println(numGuests); } } }</pre>	

Tehtävä 5: toistolause ja ehtolause (1p)

Find the Biggest (Try It Out p. 306)

Add code to the end of Listing 11-1 so that the program finds the room(s) with the largest number of guests. The program’s new output may look something like this:

4 guests in Room 1

4 guests in Room 6

To do this, have a variable named largestSoFar. Examine the array’s elements one by one and then update the value of largestSoFar whenever you find a new, record-high number.

MakeYourMarks.javaFindTheBiggest.java ×

```
1 package kt_5;
2
3 import java.util.ArrayList;
4 import java.util.List;
5
6 //Tämä luokka löytää huoneen tai huoneet, joissa on suurin vieraiden määrä.
7 public class FindTheBiggest {
8
9     // Tämä metodi on päämetodi, joka suoritetaan, kun ohjelma käynnistetään.
10    public static void main(String[] args) {
11
12        // Luo huoneiden taulukko.
13        int[] rooms = {0, 296, 342, 405, 363, 350, 323, 101};
14
15        // Luo muuttuja tallentamaan suurimman vieraiden määrän, joka on tähän mennessä nähty.
16        int largestSoFar = 0;
17
18        // Iteroi huoneiden taulukon yli ja päivitä largestSoFar aina, kun löydämme uuden ennätysmäärän.
19        for (int i = 0; i < rooms.length; i++) {
20            if (rooms[i] > largestSoFar) {
21                largestSoFar = rooms[i];
22            }
23        }
24
25        // Luo lista tallentamaan huoneiden numerot, joissa on suurin vieraiden määrä.
26        List<Integer> largestRooms = new ArrayList<>();
27
28        // Iteroi huoneiden taulukon yli uudelleen ja lisää huoneiden numerot, joissa on suurin vieraiden määrä, listaan.
29        for (int i = 0; i < rooms.length; i++) {
30            if (rooms[i] == largestSoFar) {
31                largestRooms.add(i);
32            }
33        }
34
35        // Tulosta huoneiden numerot, joissa on suurin vieraiden määrä, konsoliin.
36        for (int i = 0; i < largestRooms.size(); i++) {
37            System.out.println(largestSoFar + " guests in Room " + largestRooms.get(i));
38        }
39    }
40 }
41
42
```

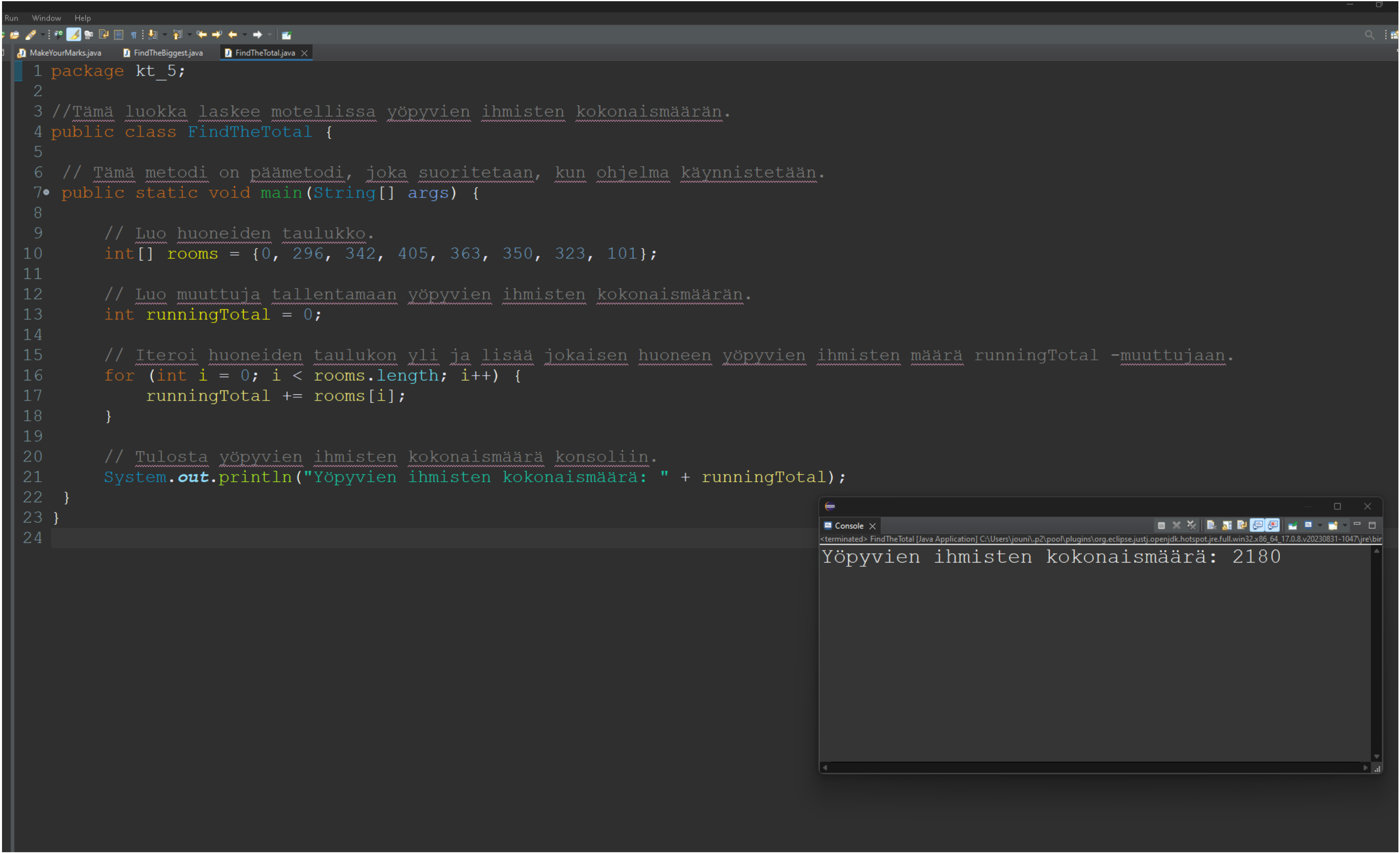
Console ×

<terminated> FindTheBiggest [Java Application] C:\Users\jouni\AppData\Local\Temp\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_17.0.8.v20230831-1047\jre\bin\java.exe
405 guests in Room 3

Tehtävä 6: toistolause (1p)

Find the Total (Try It Out p. 306)

Add code to the end of Listing 11-1 so that the program reports the total number of people staying at the motel. To do this, have a variable named runningTotal. Examine the array’s values one by one, adding each such value to the big runningTotal value.



Tehtävä 7: Selitä (3p)

Kirjassa on sivulla seuraava kommentti: "In Listing 11-4, the condition `roomNum < 10 && guests[roomNum] != 0` can be tricky. If you move things around and write `guests[roomNum] != 0 && roomNum < 10`, you can get yourself into lots of trouble.

Lue dokumentti osoitteessa: <https://users.drew.edu/bburd/JavaForDummies6/shortCircuitEval.pdf> ja selitä, miksi saatat joutua vaikeuksiin, jos tarkastelet ensin ehtoa `guests[roomNum] != 0`.

Dokumentin mukaan Javassa käytetään lyhytpiiriarviointia (short-circuit evaluation), jossa toisen ehdon testaamista voidaan välttää, jos ensimmäisen ehdon totuusarvo on jo tiedossa. Tässä tapauksessa ensimmäinen ehto on `guests[roomNum] != 0`, joka tarkistaa, onko annetun huoneen numeron vierasmäärä nollaa suurempi. Toinen ehto on `roomNum < 10`, joka tarkistaa, onko huonenumero alle 10.

Jos ehto `guests[roomNum] != 0` on tosi, ehto `roomNum < 10` ei koskaan testata. Tämä johtuu siitä, että toisen ehdon totuusarvo on jo tiedossa, ja se on tosi.

Esimerkiksi

Oletetaan, että `guests`-taulukossa on seuraavat arvot:

`guests[0] = 1`

`guests[1] = 2`

`guests[2] = 3`

`guests[3] = 4`

`guests[4] = 5`

Seuraava koodi tarkistaa, onko huoneen numero 5:ssä vieraita

`boolean hasGuests = guests[5] != 0 && roomNum < 10;`

Tämä koodi palauttaa tosi arvon, koska `guests[5] != 0` on tosi. Huonenumeroa ei koskaan testata, joten on mahdollista, että `roomNum` on suurempi tai yhtä suuri kuin 10.

Vaihtoehto

Vaihtoehtoisesti voidaan käyttää seuraavaa koodia:

`boolean hasGuests = roomNum < 10 && guests[roomNum] != 0;`

Tämä koodi testaa ensin huoneen numeron. Jos huonenumero on alle 10, se testaa sitten vieraiden määrän. Tämä estää ongelman, jossa ehto `guests[roomNum] != 0` ei koskaan testata.

Esimerkkikoodi

Seuraava koodi esittelee tämän ongelman ja sen ratkaisun:

```
import java.util.Arrays;

public class ShortCircuitEvaluation {

    public static void main(String[] args) {
        int[] guests = {1, 2, 3, 4, 5};

        // Tämä koodi voi palauttaa virheellisen arvon, jos roomNum on suurempi tai yhtä suuri kuin 10.
        boolean hasGuests = guests[5] != 0 && roomNum < 10;

        // Tämä koodi palauttaa aina oikean arvon.
        int roomNum = 5;
        hasGuests = roomNum < 10 && guests[roomNum] != 0;

        System.out.println(hasGuests);
    }
}
```

Tämä koodi tulostaa seuraavan:

```
true
```

Yhteenveto

Jos tarkastellaan ensin ehtoa `guests[roomNum] != 0`, voi joutua vaikeuksiin, jos toinen ehto `roomNum < 10` ei koskaan testata. Tämä johtuu siitä, että toisen ehdon totuusarvo on jo tiedossa, ja se on tosi.

Vaihtoehtoisesti voidaan käyttää toista järjestystä, jossa tarkastellaan ensin huoneen numeron. Tämä estää ongelman, jossa toinen ehto ei koskaan testata.

Tehtävä 8: Selitä (2p)

Käytä kirjan lukua 11 osaa ”When to close a file” hyväksesi ja selitä, miksi tiedostot on hyvä sulkea esim. `diskScanner.close()` -komennolla mahdollisimman pian lukemisen, kirjoittamisen tai muun tehtävän jälkeen?

Tiedostojen sulkeminen mahdollisimman pian on tärkeää useasta syystä.

- Resurssien vapauttaminen: Kun tiedosto suljetaan, vapautetaan järjestelmän resursseja, kuten muistia ja levytilaa. Tämä voi parantaa suorituskkyä ja vähentää järjestelmän kuormitusta.
- Virheiden välttäminen: Jos tiedostoa ei suljeta, se voi jäädä auki muistiin tai levyille. Tämä voi johtaa virheisiin, kuten tiedostovirheisiin tai muistiongelmiin.
- Turvallisuus: Avatut tiedostot voivat olla haavoittuvaisia hyökkäyksille. Sulkemalla tiedostot varmistat, että niitä ei voi käyttää haitallisesti.

Kirjan luku 11 osa ”When to close a file” antaa seuraavat ohjeet tiedostojen sulkemiseen:

- Sulje tiedosto mahdollisimman pian sen jälkeen, kun se ei enää ole käytössä.
- Sulje tiedostot aina, kun ohjelmasi päättyy.
- Sulje tiedostot, jos ohjelmasi kohtaa virheen.

Seuraavassa on esimerkki siitä, miten tiedosto voidaan sulkea `diskScanner.close()` -komennolla:

TiedostonLukija.java ×

```
1 package kt_5;
2
3 import java.io.File;
4 import java.io.FileNotFoundException;
5 import java.io.IOException;
6 import java.util.Scanner;
7
8 public class TiedostonLukija {
9
10     public static void main(String[] args) throws FileNotFoundException, IOException {
11         // Avaa tiedosto
12         File tiedosto = new File("tiedosto.txt");
13         Scanner diskScanner = new Scanner(tiedosto);
14
15         // Lue tiedosto
16         while (diskScanner.hasNextLine()) {
17             System.out.println(diskScanner.nextLine());
18         }
19
20         // Sulje tiedosto
21         diskScanner.close();
22     }
23 }
24
25
```

Console ×

<terminated> TiedostonLukija [Java Application] C:\Users\jouni\AppData\Local\Temp\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_17.0.8.v20230831-1047\jre\bin\javaw.exe (11 Oct 2023, 17:08:12 – 17:08:12) [pid: 33868]

Akku ankka asuu ankkalinssa yhdessä tupu, hupu ja lupun kanssa

Tehtävä 9: Arrayn komponenttien arvon muuttaminen, toistolause (2p)

Tally Ho! (Try It Out p. 311)

Add code to the end of Listing 11-1 so that the program reports the number of empty rooms, the number of rooms with only one guest, the number of rooms with exactly two guests, and so on. To do this, create an array named `howManyRoomsWith`. Examine the motel’s rooms one by one. When you encounter an empty room, add 1 to `howManyRoomsWith[0]`. When you encounter a room with one guest, add 1 to `howManyRoomsWith[1]`. And so on.

FileEditSourceRefactorNavigateSearchProjectRunWindowHelp

TiedostonLukija.javaRoom.java kt9.java

4

5 public class kt9 {

6

7 • public static void main(String[] args) {

8 // Luo käyttäjän syötteen lukemiseen tarkoitettu lukija.

9 Scanner lukija = new Scanner(System.in);

10

11 // Kysy motellin huoneiden lukumäärä.

12 System.out.print("Anna motellin huoneiden lukumäärä: ");

13 int huoneidenLkm = lukija.nextInt();

14

15 // Luo taulukko, joka tallentaa huoneissa olevien asukkaiden lukumäärän.

16 int[] huoneet = new int[huoneidenLkm];

17

18 // Luo asukkaiden lukumäärä jokaisesta huoneesta.

19 for (int huone = 0; huone < huoneidenLkm; huone++) {

20 System.out.print("Anna huoneen " + (huone + 1) + " asukkaiden lukumäärä: ");

21 huoneet[huone] = lukija.nextInt();

22 }

23

24 // Luo taulukko, joka tallentaa huoneiden lukumäärän jokaiselle asukkaiden

25 int[] kuinkaMontaHuonettaOn = new int[2];

26

27 // Käy motellin huoneet läpi yksitellen.

28 for (int huone = 0; huone < huoneidenLkm; huone++) {

29 // Tarkista, onko elementti taulukon rajojen sisäpuolella.

30 if (huoneet[huone] < kuinkaMontaHuonettaOn.length) {

31 // Lisää 1 vastaavaan taulukon elementtiin.

32 kuinkaMontaHuonettaOn[huoneet[huone]]++;

33 }

34 }

35

36 // Tulosta tulokset.

37 System.out.println("Tyhjiä huoneita: " + kuinkaMontaHuonettaOn[0]);

38 System.out.println("Yhden asukkaan huoneita: " + kuinkaMontaHuonettaOn[1]);

39 }

40 }

Console

<terminated> kt9 [Java Application] C:\Users\jouni\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_17.0.8.v20230831-104

Anna motellin huoneiden lukumäärä: 3

Anna huoneen 1 asukkaiden lukumäärä: 4

Anna huoneen 2 asukkaiden lukumäärä: 0

Anna huoneen 3 asukkaiden lukumäärä: 2

Tyhjiä huoneita: 1

Yhden asukkaan huoneita: 0

Tehtävä 10: array muuttuja objektissa

DOUBLE YOUR PRESSURE (Try it Out p. 318)

In Chapter 9, you create a `Student` class. Each student has a name and an ID number. For this programming challenge, imagine that each student has five grades — one for each of the five courses the student takes. Each grade is a double value from 0.0 to 4.0 (4.0 is the best). A student’s grade point average (GPA) is the average of the student’s five grade values.

In this chapter’s `Student` class, one of the fields is an array of five double values. Your program finds the student’s GPA and displays it (along with the student’s name and ID number) on the screen.