

## Ohjelmoinnin perusteet, 1. välikoe

Nimi: \_\_\_\_\_ Opiskelijanumero: \_\_\_\_\_

1.	2.	3.	4.	Yhteensä

### Ohje:

Ratkaise kaikki tehtävät. Lähdemateriaalia ja tietokonetta ei saa käyttää. Noudata ohjelmointitehtävissä Java-kielen vakiintuneita ohjelmointikäytänteitä (muuttujanimet, sisennykset jne.).

Voit olettaa kaikissa tehtävissä, että syötteen lukemiseen tarvittava kurssikirjan mukainen **Oma**-luokka on käytössäsi. Luokka tarjoaa seuraavat metodit:

- **Oma.lueInt()** lukee ja palauttaa kokonaisluvun
- **Oma.lueDouble()** lukee palauttaa liukuluvun (desimaaliluvun)
- **Oma.lueString()** lukee ja palauttaa merkkijonon
- **Oma.lueChar()** lukee ja palauttaa merkin

Lisäksi:

- Voit käyttää neliöjuuren laskemiseksi metodia **Math.sqrt**, jolle annetaan parametrina **double**-tyyppinen juurettava. Metodi palauttaa neliöjuuren arvon **double**-tyyppisenä. Esimerkiksi **Math.sqrt(2.0)** palauttaa arvon **1.4142135623730951**.
- Voit laskea potenssiinkorotuksen käyttämällä metodia **Math.pow**, joka saa parametreina kantaluvun ja eksponentin. Metodi palauttaa potenssiinkorotuksen tuloksen **double**-tyyppisenä. Esimerkiksi **Math.pow(2.5, 2)** palauttaa arvon **6.25**.

1. Tutustu alla olevaan ohjelmaan ja ratkaise tehtävät. (Kukin kohta 1 p.)

```
public class Ympyrä {
    public static void main (String[] args) {

        _____ i;
        _____ säde;
        _____ ala;

        for (i = 1; i<=3; i++) {
            System.out.println("Anna ympyrän ala.");
            ala = Oma.lueDouble();
            säde = Math.sqrt(ala/Math.PI);
            System.out.println("Ympyrän säde on " + säde);
        }
    }
}
```

- a) Kerro selkeästi ja tarkasti ohjelman idea: mitä ohjelma tekee?

---

---

- b) Täydennä viivoille määrittelylauseista puuttuvat tietotyypit tarkoituksenmukaisella tavalla.
- c) Oletetaan, että et halua käyttää `Math`-luokassa määriteltyä vakiota `Math.PI` vaan korvata sen itse määrittelemälläsi vakiolla `PII`, jolle annat arvoksi 3.1415927. Tee ohjelmakoodiin selkeästi tarvittavat muutokset.
- d) Ohjelmassa käytetään muuttujanimiä `säde`, `ala` ja `i`. Yliviivaa seuraavasta luettelosta sellaiset muuttujanimet, jotka olisivat syntaktisesti virheellisiä eli joita Java-kääntäjä ei hyväksyisi:

Ala    12kk    Ångström\_127897    Kehän.neliö    henkka&maukka  
SÄDE2    s\$    SÄDE

- e) Yliviivaa seuraavasta luettelosta sellaiset muuttujanimet, jotka olisivat isojen ja pienten kirjainten käytön puolesta Java-kielen vakiintuneen nimeämiskäytännön vastaisia muuttujanimen `ala` korvaajina em. ohjelmassa:

a    A    ALA    Ala    AnnettuAla    Annettuala    annettuAla  
annettuala

- f) Ohjelmassa esiintyy lause `i++`. Kirjoita lauseelle kolme muuta esitystapaa, jotka toimisivat tuossa ohjelman kohdassa täsmälleen samalla tavalla kuin edellä mainittu lause.

---

---

---



3. Kirjoita ohjelma, joka kysyy käyttäjältä massan keskiaikaisten mittojen mukaisesti: leivisköinä, nauloina ja luoteina. Ohjelma muuntaa syötteen täysiksi kilogrammoiksi ja grammoiksi ja ilmoittaa tuloksen käyttäjälle.

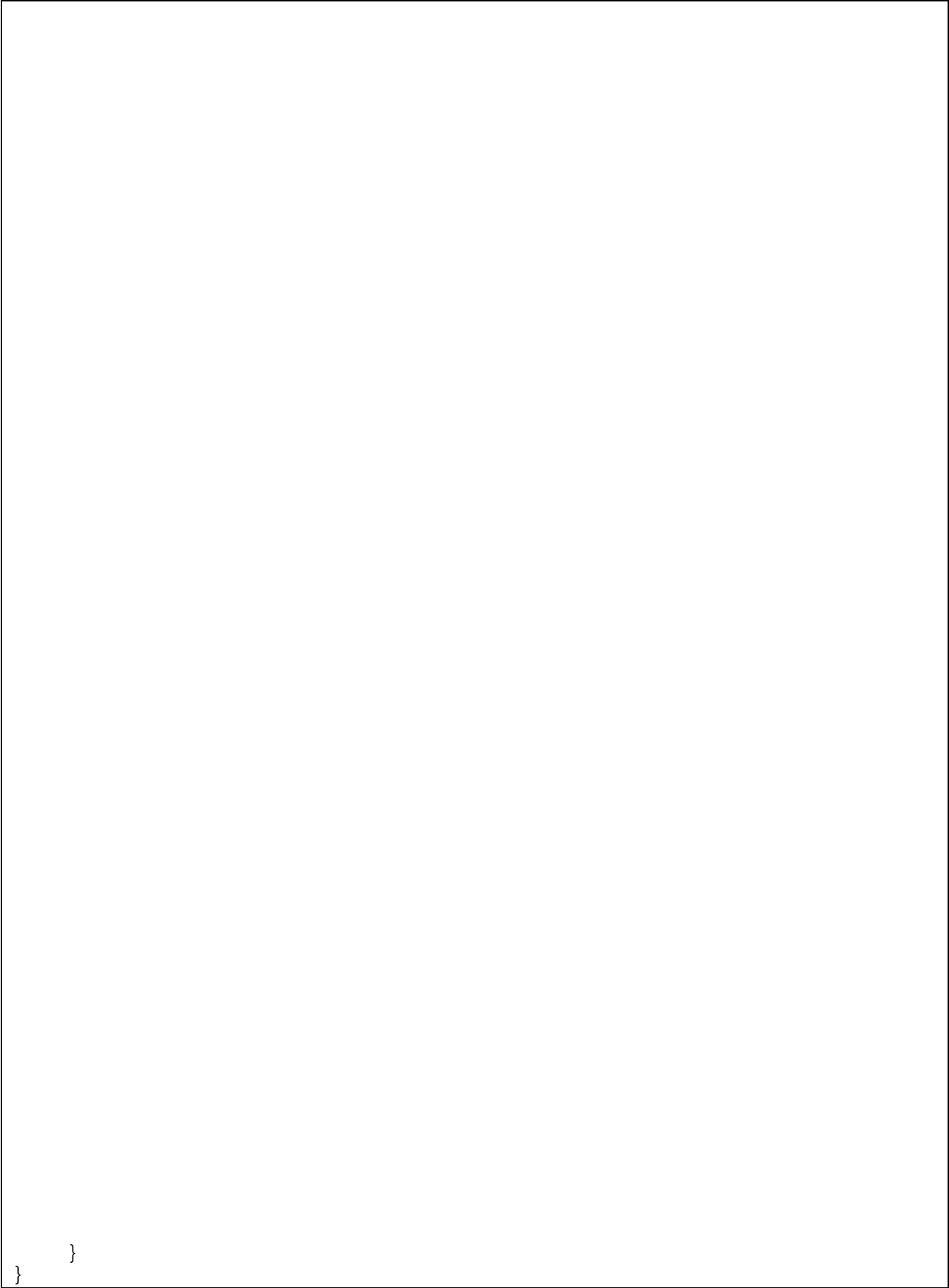
Sekä leiviskät, naulat että luodit saavat olla desimaalilukuja. Tuloksessa ilmoitetaan aina täydet kilogrammat kokonaislukuna sekä yli jäävät grammat desimaaleineen.

Muunnossääntö: *Yksi leiviskä on 20 naulaa. Yksi naula on 32 luotia. Yksi luoti on 13,3 grammaa.*

Ohjelman on toimittava alla olevan esimerkin mukaisesti. (Esimerkissä 3 leiviskää, 9 naulaa ja 13,5 luotia on luoteina 2221,5. Tämä on grammoina  $2221,5 \times 13,3 = 29545,95$ , joka ilmoitetaan muodossa 29 kilogrammaa ja 545,95 grammaa). Virheellisiin syötteisiin ei tarvitse varautua.

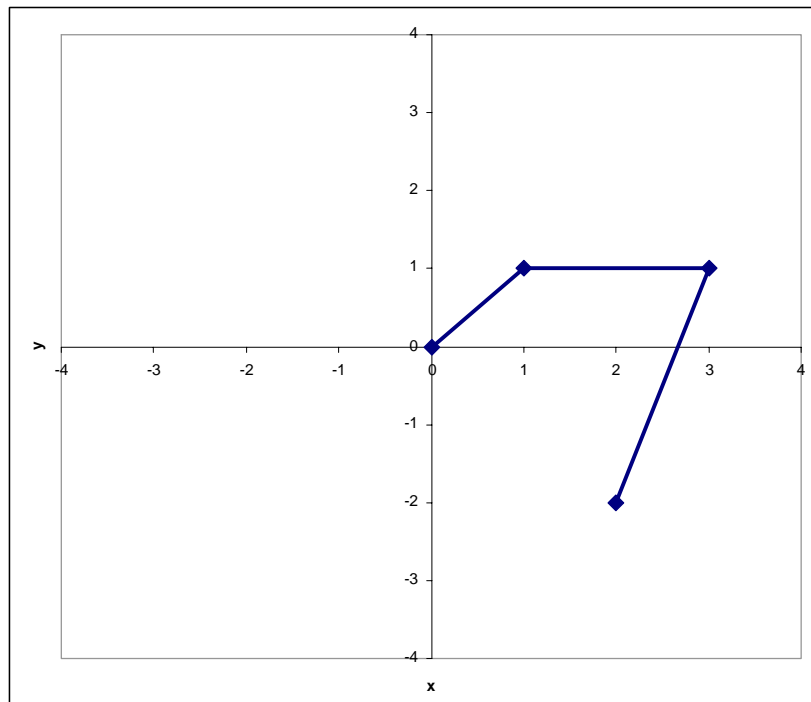
```
Anna leiviskät.  
3  
Anna naulat.  
9  
Anna luodit.  
13.5  
29 kilogrammaa ja 545.95 grammaa.
```

```
public class Ratkaisu {  
  
    public static void main (String[] args) {
```



4. Kirjoita ohjelma, joka kysyy käyttäjältä reittipisteitä tason suorakulmaisessa koordinaatistossa ja tulostaa lopussa reittipisteiden määrän ja kokonaismatkan. Lähtö tapahtuu aina origosta.

Esimerkki: käyttäjä haluaa syöttää kolme reittipistettä:  $(x=1, y=1)$ ,  $(x=3, y=1)$  ja  $(x=2, y=-2)$ . Tätä vastaa kuvan tilanne:



Ohjelma toimii näin:

Anna seuraavan pisteen koordinaatit.

x:

1

y:

1

Jatketaanko (k/e)?

k

Anna seuraavan pisteen koordinaatit.

x:

3

y:

1

Jatketaanko (k/e)?

k

Anna seuraavan pisteen koordinaatit.

x:

2

y:

-2

Jatketaanko (k/e)?

e

Reittipisteitä 4, matka yht. 6.576491222541474

Huomaa, että ensimmäinen reittipiste on aina origo ( $x=0, y=0$ ) eikä sitä kysytä käyttäjältä.

Kahden reittipisteen  $(x_1, y_1)$  ja  $(x_2, y_2)$  välinen matka saadaan lausekkeesta

$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$ . Lue ohjesivun vihjeet.

Reittipiste kysytään käyttäjältä siten, että sen x- ja y-koordinaatti kysytään erikseen. Kunkin pisteen syötön jälkeen käyttäjältä kysytään, haluaako tämä jatkaa. Vaihtoehdot ovat **k** ja **e**. Jos käyttäjä vastaa jotain muuta kuin **k** tai **e**, vastaus pyydetään uudelleen. Jos vastaus oli **k**, kysytään seuraava reittipiste. Muussa tapauksessa tulostetaan lopputulokset ja ohjelman suoritus päättyy.

```
public class Ratkaisu {  
  
    public static void main (String[] args) {
```

