Ohjelmointiharjoitus 10

# 1. Selvitä JDK:n dokumentaatiosta, mitä luokalla java.lang.Character voi tehdä?

Luokka java.lang.Character sisältää useita staattisia metodeja, jotka mahdollistavat merkkijonojen käsittelyn. Esimerkiksi:

* isLetter(char ch) - palauttaa totuusarvon, joka kertoo, onko annettu merkki kirjain vai ei.
* isDigit(char ch) - palauttaa totuusarvon, joka kertoo, onko annettu merkki numero vai ei.
* toUpperCase(char ch) - palauttaa annetun merkin isoilla kirjaimilla kirjoitettuna.
* toLowerCase(char ch) - palauttaa annetun merkin pienillä kirjaimilla kirjoitettuna.

# 2. Entä mitä tekee java.lang.Math-luokan metodi abs?

java.lang.Math-luokan abs-metodi palauttaa annetun numeron absoluuttisen arvon (eli sen etäisyyden nollasta). Esimerkiksi:

int x = -5;

int absX = Math.abs(x); // absX:n arvo on 5

# 3. Laadi ohjelma, jossa käyttäjän syöttämä pituus (cm) ilmoitetaan jalkoina (ft) ja tuumina (in). Toteuta muunnos myös toisinpäin. Tee ainakin muunnoksille omat aliohjelmat.

Text

Description automatically generated

# 4. Mitä Javan dokumentaatio sanoo omien ohjelmien kommentoinnista luokkien, metodien, muuttujien osalta?

Javan dokumentaatio suosittelee, että ohjelmakoodi kommentoidaan huolellisesti kahdentyyppisillä kommenteilla: peruskommenteilla (kaksoisviiva //) ja javadoc-kommenteilla (kahdella tähdellä /\*\*). Javadoc-kommentit ovat erityistyyppisiä kommentteja, jotka sijoitetaan luokan, metodin tai muuttujan yläpuolelle ja niitä käytetään automaattisesti luotavien dokumentaatioiden luomiseen. Selkeiden ja ymmärrettävien kommenttien käyttö auttaa muita kehittäjiä ymmärtämään koodin toimintaa ja tarkoitusta sekä helpottaa koodin ylläpitoa ja muokkaamista.

* Luokan tarkoitus ja käyttötarkoitus
* Metodin parametrit ja paluuarvo
* Muuttujan käyttötarkoitus ja ominaisuudet

Yleisesti ottaen, Javan dokumentaatio suosittelee, että ohjelmakoodi kommentoidaan selkeästi ja ymmärrettävästi, jotta muut kehittäjät voivat helposti ymmärtää koodin toimintaa ja tarkoitusta. Kommenttien avulla on helpompi ylläpitää ja muokata koodia, mikä voi säästää aikaa ja vaivaa pitkällä aikavälillä.

# 5. Toteuta edellinen tehtävässä 3.

sama koodi kun 3 tehtävässä mutta kommentit vaihdettu vastaamana javadocia

Text

Description automatically generated

# 6. Luo Javadoc-dokumentti

kuva siitä kuntein javadocin ja myös kuva javadocista

Graphical user interface, text

Description automatically generated

# 7. Selvitä, miten java-ohjelman voi ajaa komentoriviltä.

Java-ohjelman ajaminen komentoriviltä on yksinkertaista. Voit seurata seuraavia vaiheita ohjelman ajamiseksi komentoriviltä:

1. Avaa komentokehote tai terminaali.
2. Siirry komentokehotteessa tai terminaalissa Java-projektin juurihakemistoon.
3. Käännä Java-ohjelma komennolla javac ja anna sen jälkeen argumentiksi .java-tiedoston polku. Esimerkiksi, jos ohjelma on tiedostossa MyProgram.java, komento olisi javac MyProgram.java.
4. Kun kompilaatio on valmis, voit ajaa ohjelman komennolla java ja anna sen jälkeen argumentiksi ohjelman luokan nimen, joka sisältää main-metodin. Esimerkiksi, jos ohjelman main-metodi on luokassa MyProgram, komento olisi java MyProgram.
5. Ohjelma suoritetaan ja tulostaa mahdollisen syötteen, laskelman tai muun ohjelman toiminnan komentoriville.

On tärkeää huomata, että ohjelman ajamiseen tarvitaan JDK:n (Java Development Kit) asennus. Voit asentaa JDK:n Oracle-sivustolta löytyvän ohjelmiston avulla.

# 7. Selvitä, miten java-ohjelman voi ajaa komentoriviltä.

Java-ohjelman ajaminen komentoriviltä on yksinkertaista. Voit seurata seuraavia vaiheita ohjelman ajamiseksi komentoriviltä:

1. Avaa komentokehote tai terminaali.
2. Siirry komentokehotteessa tai terminaalissa Java-projektin juurihakemistoon.
3. Käännä Java-ohjelma komennolla javac ja anna sen jälkeen argumentiksi .java-tiedoston polku. Esimerkiksi, jos ohjelma on tiedostossa MyProgram.java, komento olisi javac MyProgram.java.
4. Kun kompilaatio on valmis, voit ajaa ohjelman komennolla java ja anna sen jälkeen argumentiksi ohjelman luokan nimen, joka sisältää main-metodin. Esimerkiksi, jos ohjelman main-metodi on luokassa MyProgram, komento olisi java MyProgram.
5. Ohjelma suoritetaan ja tulostaa mahdollisen syötteen, laskelman tai muun ohjelman toiminnan komentoriville.

On tärkeää huomata, että ohjelman ajamiseen tarvitaan JDK:n (Java Development Kit) asennus. Voit asentaa JDK:n Oracle-sivustolta löytyvän ohjelmiston avulla.

# 8. Näytä edellinen käytännössä ohjelmalla, joka laskee kaksi lukua yhteen. Tee versio, jossa luvut luodaan ohjelman sisällä sekä versio, jossa luvut annetaan komentorivillä.

Graphical user interface, text

Description automatically generated

# 9. Alla on kuvakaappaus luokasta Test. Mitä ohjelma tekee? Poista koodista breakohjelman toimintaa muuttamatta.

Text

Description automatically generated

Tulostaa taulukon numbers alkioita kunnes saavuttaa ensimmäisen luvun, joka on yhtä suuri kuin30. Tämän jälkeen se lopettaa for-silmukan suorittamisen break-käskyllä.

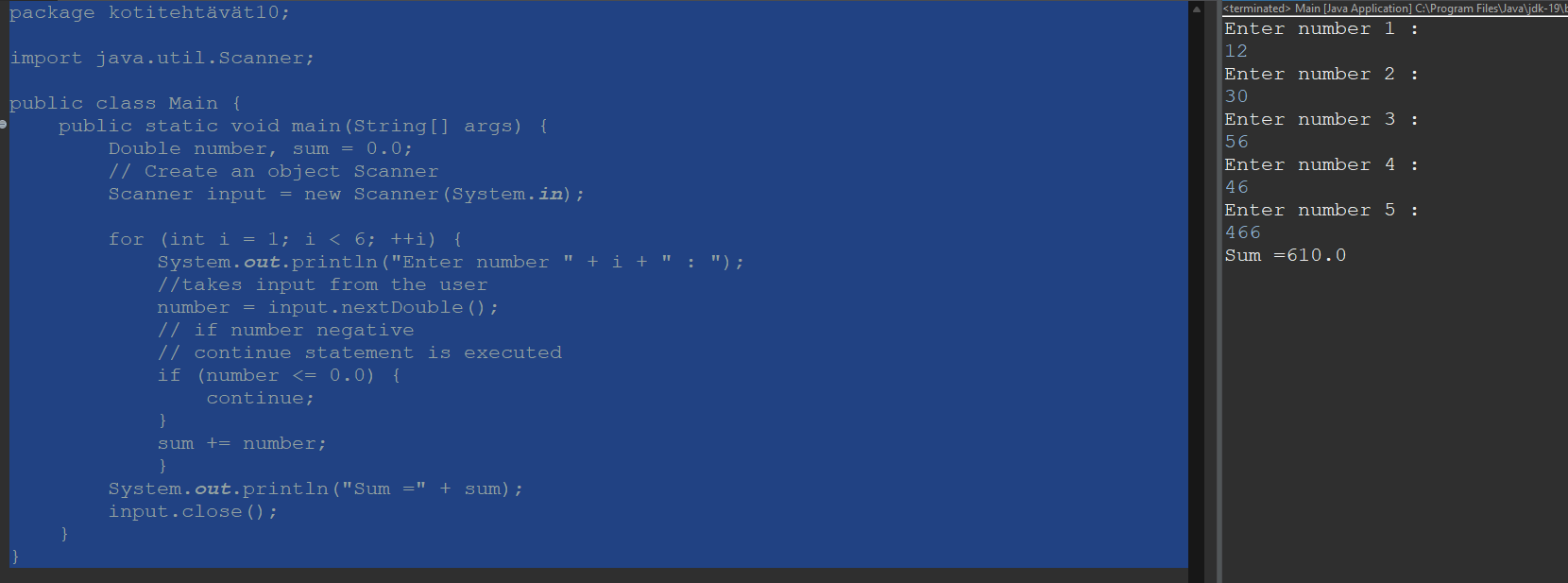
Jos break poistetaan ohjelmasta muuttamatta sen toimintaa se voidaan poistaa ja jättää muuten koodi ennalleen tuolloin for-silmukka suoritetaan loppuun asti ja tulostetaan kaikki taulukon numbers alkiot

Alla on koodi missä koodi ilman break-toimintoa

Text

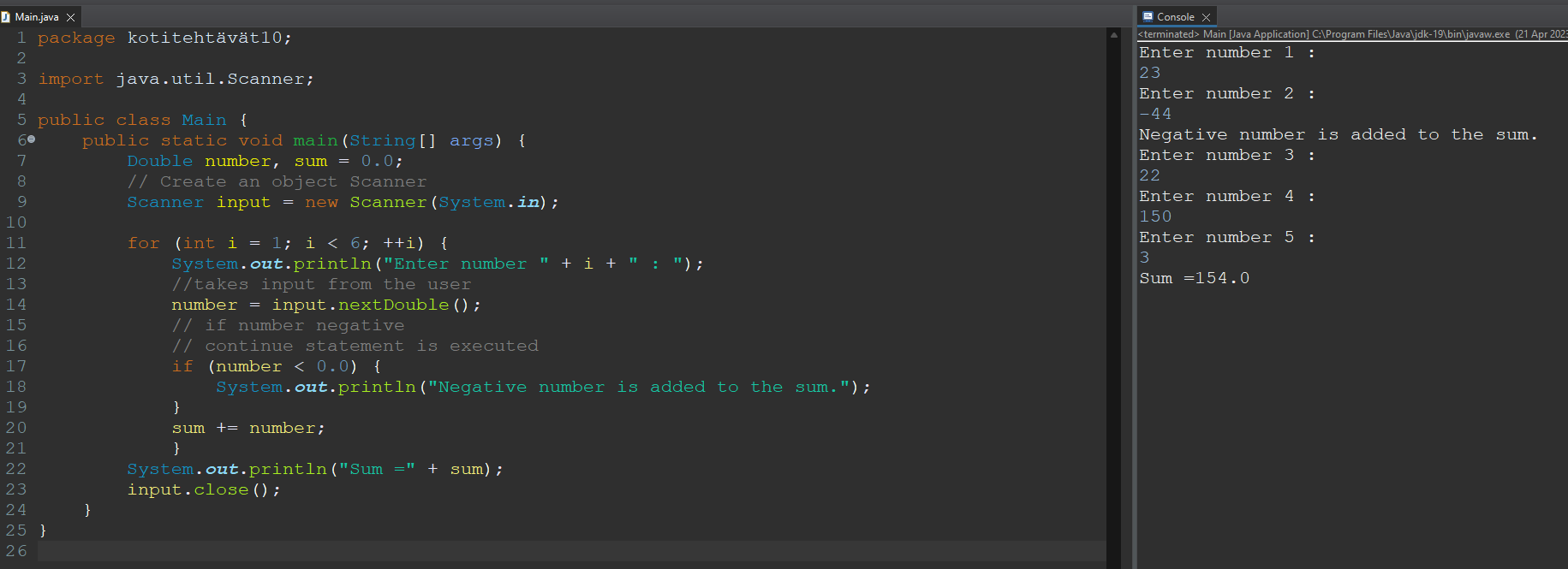
Description automatically generated

# 9. Alla on kuvakaappaus luokasta Test. Mitä ohjelma tekee? Poista koodista breakohjelman toimintaa muuttamatta.



Pyytää käyttäjältä syöttämään viisi lukua jokainen syötetään erikseen, jos syötetty luku on negatiivinen se ohitetaan continue komennon avulla ja siirtyy seuraavaa annettavaan lukuun. Muussa tapauksessa ohjelma lisää syötetyn luvun summaan ja lopuksi ohjelma tulostaa summattujen lukujen tuloksen

Mikäli poistaa continue-komennon, voisi sen tilalle käyttää if-lauseen sisälle toista if-lausetta, joka tarkistaa onko syötetty luku negatiivinen. Tällöin ohjelma laskee myös negatiiviset luvut mukaan summaan. Seuraava koodi esittelee muutetun ohjelman:



Ohjelma ottaa muutoksien negatiiviset luvut mukaan summaan mutta tulostaa samalla ilmoituksen että negatiivinen luku lisätty summaan