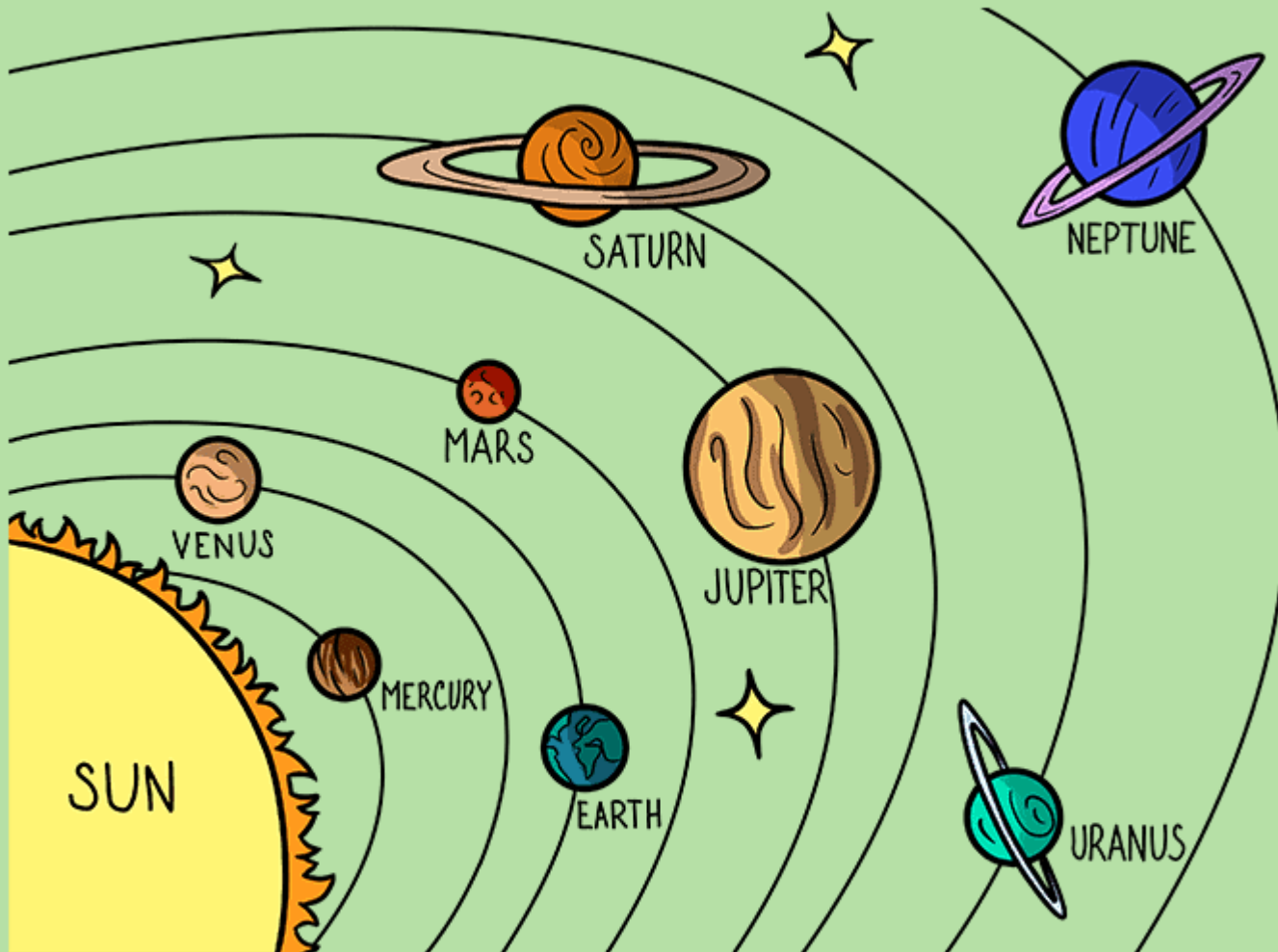


Planeettafakta

Level 3 - Java



Johdanto



Aurinkokuntamme muodostavat aurinko ja kaikki sen painovoimakenttään sidotut taivaankappaleet: planeetat Merkurius, Venus, Maa, Mars, Jupiter, Saturnus, Uranus ja Neptunus sekä kääpiöplaneetta Pluto. Lisäksi aurinkokuntamme koostuu kymmenistä kuista, miljoonista asteroideista, komeetoista ja meteoreista.

Nykyaikainen avaruustutkimus ei olisi mahdollista ilman ohjelmointia ja tietojenkäsittelytiedettä.

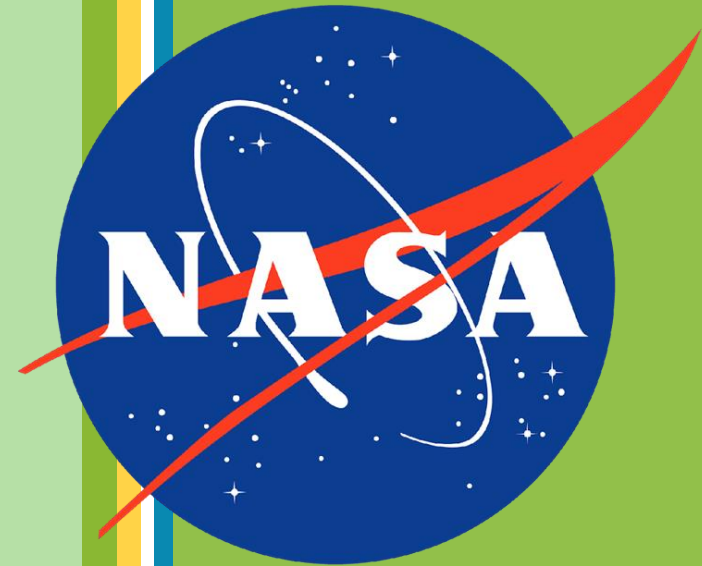
NASA kehittää parhaillaan avaruusrobotteja, jotka käyttävät tekoälyä aurinkokunnan haiki matkaamiseen.

Tulevaisuus on tähdissä!

Tehtävä

NASA and NASA ja ESA etsivät lisää työntekijöitä avaruushjelmiinsa, mutta heillä on ongelma: uudet työntekijät ovat ohjelmoijia, eivät fyysikoita!

NASA on antanut sinulle tehtävän luoda heille planeettoja käsittelevä tietokanta, josta ohjelmoijat oppisivat lisää alasta.



Prosessi

Projektissa luomme planeetoista kertovan tiedoston. Opit esimerkiksi:

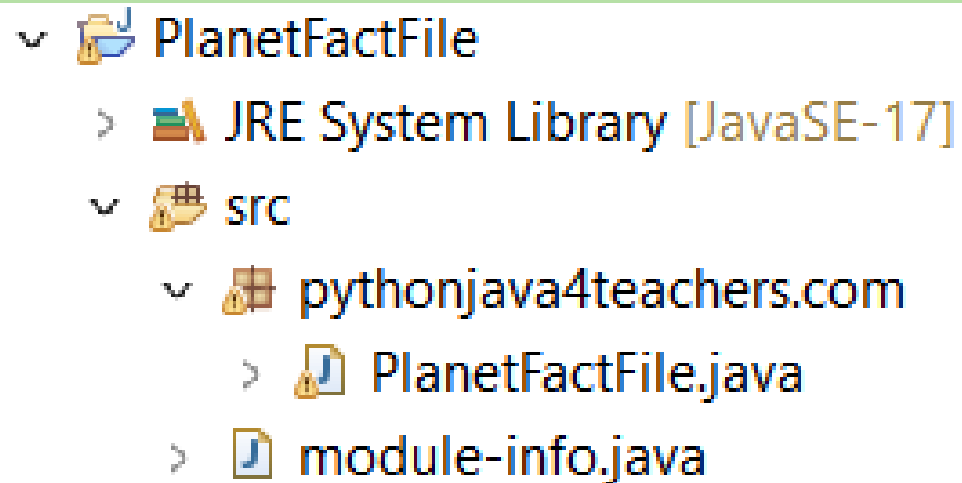
- Keräämään tietoa aurinkokunnastamme
- Miten käyttäjälle annetaan mahdollisuus valita planeetta, josta he haluavat oppia lisää.
- Miten tietoa saadaan ulkoisesta tiedostosta ja näyttää käyttäjälle faktatiedoston valitusta planeetasta.



Ennen aloitusta...

Aloita luomalla uusi Java-projekti. Jos et ole avian Varma, miten se tehdään, katso Johdatus Javaan -tiedostoa.

Käynnistä IDE seuraavan näköiseksi:



Virheilmoitukset

Java on aavistuksen monimutkaisempi kuin Python-kieli. Kuitenkin, monimutkaisuus tuo lisää mahdollisuuksia sekä erilaisia virheilmoituksia.

Aseta hiiri punaisella alleviivatun koodinosan päälle, niin Eclipse IDE näyttää erilaisia ratkaisuvaihtoehtoja.

```
import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.Font;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.Polygon;

import javax.swing.JFrame;
import javax.swing.SwingUtilities;

public class RoadSi
{
    public void paint()
    {
        super.paint();

        Polygon b = new Polygon();
        for (int i = 0; i < 10; i++)
        {
            b.addPoint(i * 10, i * 10);
        }

        g.setColor(Color.BLUE);
        g.fillPolygon(b);
        g.drawPolygon(b);
    }
}
```

The type javax.swing.SwingUtilities is not accessible

7 quick fixes available:

- Create class 'SwingUtilities' in package 'javax.swing'
- Create record 'SwingUtilities' in package 'javax.swing'
- Create interface 'SwingUtilities' in package 'javax.swing'
- Create annotation 'SwingUtilities' in package 'javax.swing'
- Create enum 'SwingUtilities' in package 'javax.swing'
- Add 'requires java.desktop' to module-info.java
- Fix project setup...

Esimerkiksi, jos saat virheilmoituksen 'this package is not accessible' (tätä pakettia ei löydy), valitse jokin näistä:

1. askel

Päämetodi

Kun olet tehnyt luokan, kirjoita seuraavat asiat:

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

Tarvitsemme main()-metodin. Main- eli päämetodi on koodin alkupiste. Jokainen Javaprojekti tarvitsee main()-metodin.

Silmukat

Silmukka on koodia, jota toistetaan, kunnes silmukalle annettu ehto toteutuu.

Java-kielessä on kaksi perussilmukkaa; FOR-silmukat ja WHILE-silmukat

WHILE-silmukkaa toistetaan, kunnes sille annettu ehto on epätosi, FALSE.

```
int num = 0;
while (num < 5) {
    num ++;
    System.out.println("This loop will repeat 5 times!");
}
```

```
while (true) {
    System.out.println("This is an infinite loop!");
}
```

FOR-silmukkaa toistetaan tietyn lukumäärän ajan.

```
for (int i = 0; i<5; i++) {
    System.out.println("This loop will repeat 5 times!");
}
```

2. askel

Valikon tekeminen

Käyttäjä saa valita, mitä planeettaa hän haluaa tutkia. Tätä varten teemme alkuvalikon.

```
public static void main(String[] args) {  
  
    while (true) {  
  
        System.out.println("\n\n --- PLANET FACT FILE ---");  
        System.out.println("1) Mercury");  
        System.out.println("2) Venus");  
        System.out.println("3) Earth");  
        System.out.println("4) Mars");  
        System.out.println("5) Jupiter");  
        System.out.println("6) Saturn");  
        System.out.println("7) Uranus");  
        System.out.println("8) Neptune");  
  
    }  
  
}
```

Pääohjelma (main class) tulostaa listan planeetoista, jotka meillä on tietokannassa. Tämä laitetaan silmukkaan, jotta se toistuu ja toistuu.

Kun ajat koodisi, se tulostaa listaa jatkuvasti. Se pysäytetään odottamalla käyttäjältä numeroa.

3. askel

Käyttäjän valinta

```
public static void main(String[] args) {  
    while (true) {  
        String planetFile;  
  
        System.out.println("\n\n --- PLANET FACT FILE ---");  
        System.out.println("1) Mercury");  
        System.out.println("2) Venus");  
        System.out.println("3) Earth");  
        System.out.println("4) Mars");  
        System.out.println("5) Jupiter");  
        System.out.println("6) Saturn");  
        System.out.println("7) Uranus");  
        System.out.println("8) Neptune");  
  
        Scanner userInputObj = new Scanner(System.in);  
        System.out.println("Enter the name of the planet: ");  
        String planet = userInputObj.nextLine();  
        planetFile = planet.toLowerCase() + ".txt";  
    }  
}
```

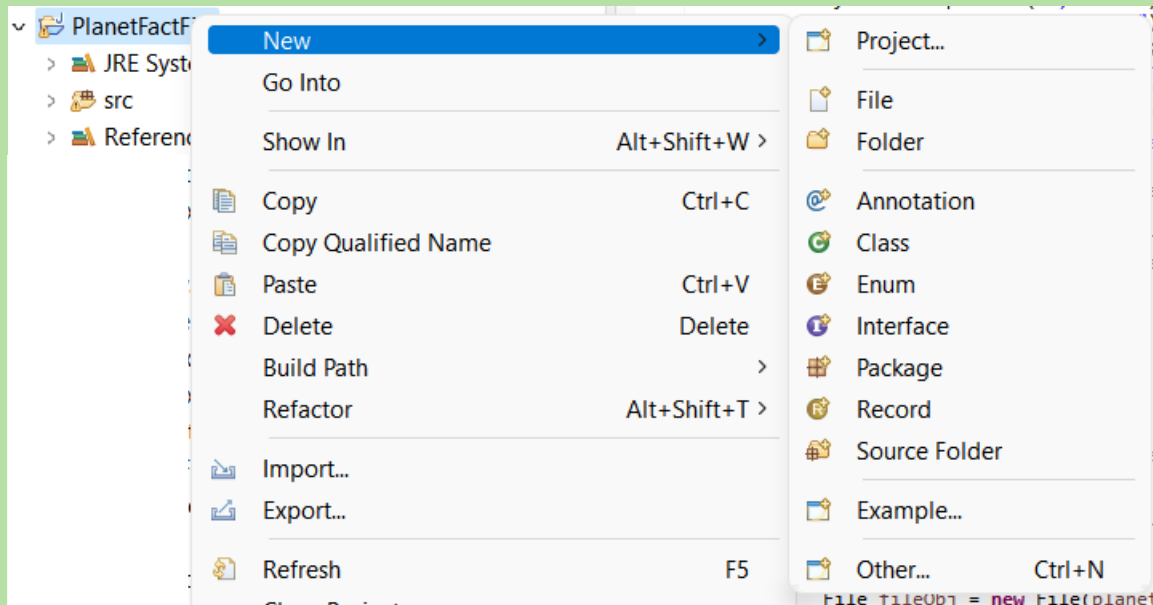
Java-kielessä käyttäjän syöte saadaan käyttämällä skanneria (Scanner object). Skanneri osaa lukea ja tallettaa käyttäjän antamat näppäinkomennot.

Oheinen koodi pyytää käyttäjää antamaan sen planeetan nimi, josta hän haluaa lisätietoa.

Se tallettaa nimen planet-nimiseen muuttujaan ja tekee siitä tiedoston nimen.

4. askel

Luo tiedostoja



Talletamme tiedot planeetoista eri tiedostoihin.

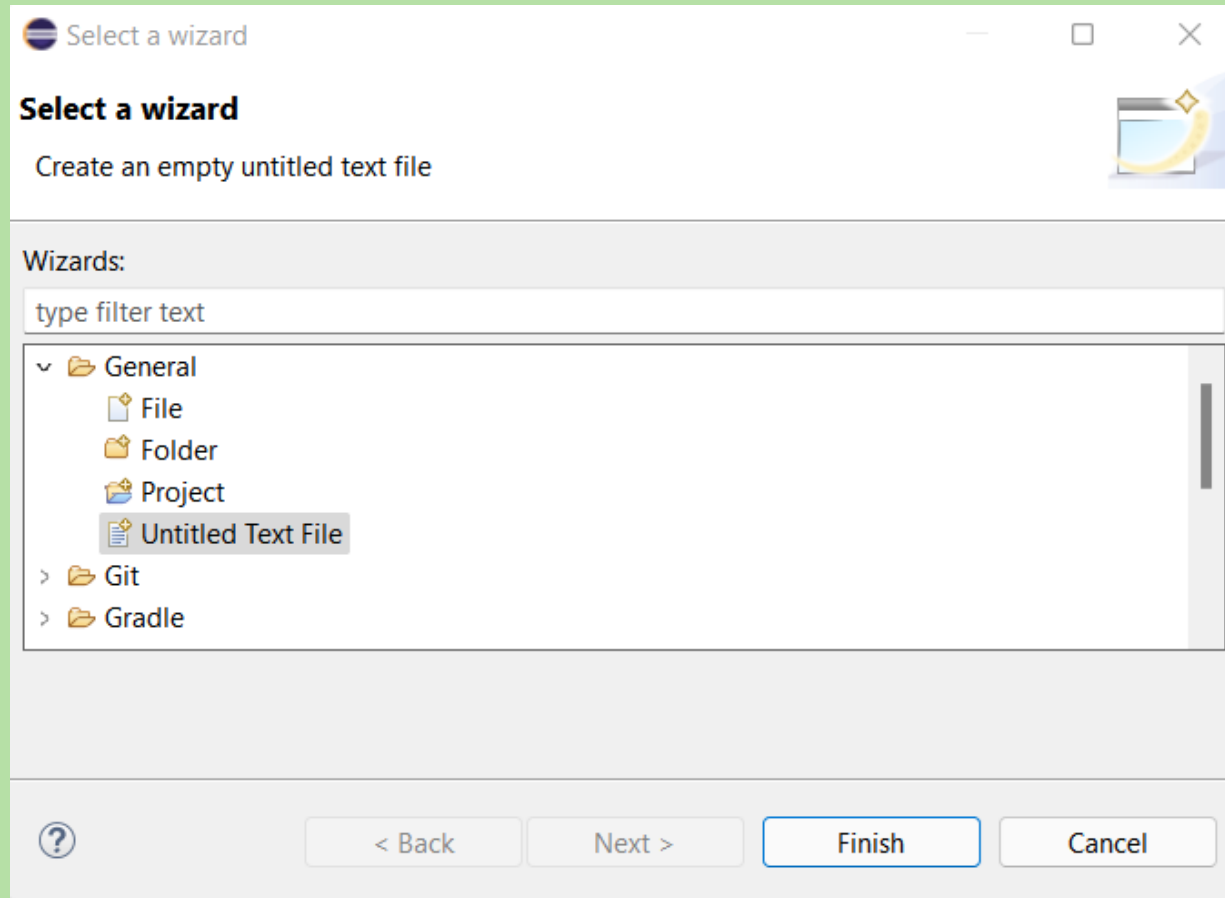
Tehdään ensin ne tiedostot ja vasta sitten luetaan tiedostoja Java-kielillä.

Klikkaa oikealla napilla Project Explorer -kohtaa.

Valitse New > Other...

5. askel

Tee tiedostot



Valitse 'Untitled Text File' ja paina Finish.



Aliohjelmat (subroutines)

Aliohjelmat ovat sellaisia osia ohjelmasta, joita toistetaan useasti ohjelmassasi.

```
public static void greeting() {  
    System.out.println("Hello World!");  
}
```

Aliohjelmiin voi “tallettaa” ohjelmakoodia, ja se koodi suoritetaan vain kun aliohjelmaa kutsutaan.

Java-kielessä on kahta päätyyppiä aliohjelmia: proseduurit ja funktiot.

Proseduurit eivät palauta kutsujalle mitään, mutta funktio palauttaa kutsujalle jonkun arvon.

Käyttämällä aliohjelmia pystyt tekemään paremmin ja helpommin ylläpidettävää koodia ja ohjelmasi on paremmin järjestetty, organisoitu ja paljon ymmärrettävämpi.

Parametrit ja argumentit

Aliohjelman määritelmässä on sulkuihin merkitty parametrit. Määritelmävaiheessa niillä ei ole arvoa, mutta kun sitä kutsutaan, kutsussa pitää olla arvot, joita aliohjelma osaa hyödyntää.

```
public static void addition(int num1, int num2) {  
    System.out.println(num1 + num2);  
}
```

Aliohjelman kutsussa olevia arvoja sanotaan argumenteiksi.

6. askel

Tutki ja listaa

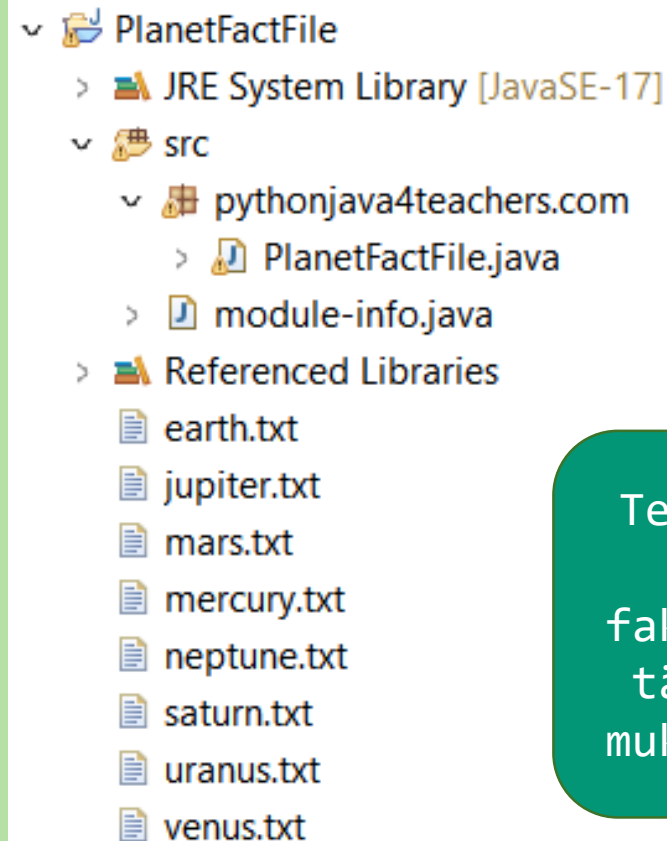
Jokaiseen tekstitiedostoon kirjoitetaan faktoja planeetasta.

Alla on muutama fakta maapallosta.

```
planetName: The Earth
radius: 6371 km
orbitalPeriod: 365 days
distanceToSun: 149.6 million km
surfaceArea: 510.1 million km^2
lengthOfDay: 24 hours
density: 5.51 g/cm^3
description: Earth is the planet we live on!
Currently the only known place in the universe to
support life.
funFact: 70% of the Earth's surface is covered in
water.
```

Etsi mielenkiintoista tietoa. Tee jokaisesta 8 planeetasta oma tiedosto ja varmista, että ne ovat oikeasti olemassa.

Project Explorer -ikkunan pitäisi näyttää alla olevalta.



Tekstitiedostot, joissa on faktatiedot, ovat tässä projektin mukana koko ajan.

7. askel

Lukeminen tekstitiedostosta

```
public static void main(String[] args) {  
    while (true) {  
        String planetFile;  
  
        System.out.println("\n\n --- PLANET FACT FILE ---");  
        System.out.println("1) Mercury");  
        System.out.println("2) Venus");  
        System.out.println("3) Earth");  
        System.out.println("4) Mars");  
        System.out.println("5) Jupiter");  
        System.out.println("6) Saturn");  
        System.out.println("7) Uranus");  
        System.out.println("8) Neptune");  
  
        Scanner userInputObj = new Scanner(System.in);  
        System.out.println("Enter the name of the planet: ");  
        String planet = userInputObj.nextLine();  
        planetFile = planet.toLowerCase() + ".txt";  
    }  
}  
  
public static void factFile(String planetFile) {  
  
}
```

Pääohjelman main()-ulkopuolelle tehdään aliohjelma. Se ottaa parametrina sisäänsä planeettatiedoston nimen ja tulostaa ruudulle tietoja planeetasta.

8. askel

Lukeminen tekstitiedostosta

```
public static void factFile(String planetFile) {  
    Map<String, String> dictionary = new HashMap<String, String>();  
  
    try {  
        File fileObj = new File(planetFile);  
        Scanner scannerObj = new Scanner(fileObj);  
  
    } catch (FileNotFoundException e) {  
        System.out.println("An error occurred");  
    }  
  
}
```

Käytämme TRY CATCH -menetelmää, koska aina lukeminen ei onnistu. Tiedosto voi olla kadonnut tai hajonnut.

Tiedoston sisältämät tiedot talletetaan dictionary:yn.

Tässäkin hyödynnämme skanneri-objektia lukemisessa.

9. askel

Lukeminen tiedostosta

```
try {
    File fileObj = new File(planetFile);
    Scanner scannerObj = new Scanner(fileObj);

    while (scannerObj.hasNextLine()) {
        String data = scannerObj.nextLine();
        //System.out.println(data);
        String[] tempLine = data.split(": ");
        dictionary.put(tempLine[0],tempLine[1]);
    }
    scannerObj.close();
    String planetName = dictionary.get("planetName");
    String radius = dictionary.get("radius");
    String orbitalPeriod = dictionary.get("orbitalPeriod");
    String distanceToSun = dictionary.get("distanceToSun");
    String surfaceArea = dictionary.get("surfaceArea");
    String lengthOfDay = dictionary.get("lengthOfDay");
    String density = dictionary.get("density");
    String description = dictionary.get("description");
    String funFact = dictionary.get("funFact");
}
```

WHILE-silmukalla käydään läpi tiedoston jokainen rivi ja se talletetaan dictionaryyn.

Suljetaan skanneri kun kaikki rivit on luettu ja talletetaan faktat omiin muuttujiinsa.

Tekstitiedostossa tiedot voivat olla missä tahansa järjestyksessä, mutta tämä löytää ne silti.

10. askel

Tulosta faktat

Tulostetaan informaatio rudulle käyttämällä PRINTLN-käskyjä.

```
String description = dictionary.get("description");
String funFact = dictionary.get("funFact");

// Display Fact File

System.out.println("\n --- " + planetName + " --- \n");
System.out.println("# " + description + "\n");

System.out.println(" --- How far from the sun? ---");
System.out.println("# " + planetName + " is " + distanceToSun + " from the sun.\n");

System.out.println(" --- Boring Maths Info --- ");
System.out.println("# It has a surface area of " + surfaceArea);
System.out.println("# It has a radius of " + radius);
System.out.println("# It has a density of " + density);

System.out.println("\n --- Extra Info ---");
System.out.println("# The days are " + lengthOfDay + " long. 1 year is the same as " + orbitalPeriod);
System.out.println("# " + funFact);

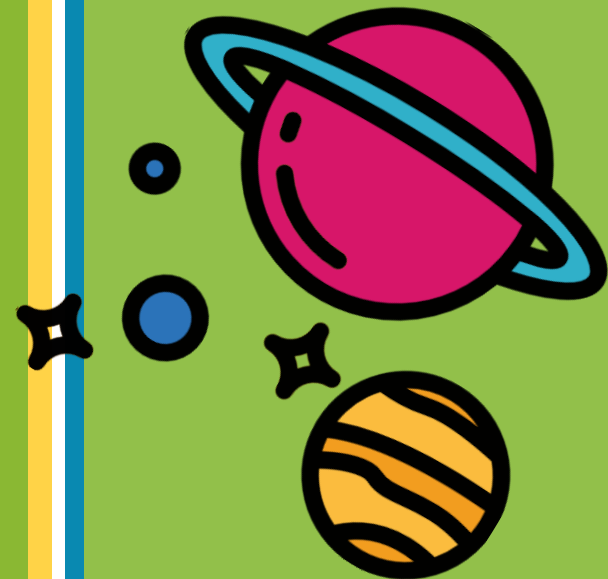
Scanner pressEnter = new Scanner(System.in);
System.out.println("\n-- Press enter to continue --");
pressEnter.nextLine();
```

Muokkaa tätä kohtaa koodista, jotta saat tulostuksen kauniiksi ja nätiksi.

11. askel

Kutsu aliohjelmaa

```
public static void main(String[] args) {  
  
    while (true) {  
  
        String planetFile;  
  
        System.out.println("\n\n --- PLANET FACT FILE ---");  
        System.out.println("1) Mercury");  
        System.out.println("2) Venus");  
        System.out.println("3) Earth");  
        System.out.println("4) Mars");  
        System.out.println("5) Jupiter");  
        System.out.println("6) Saturn");  
        System.out.println("7) Uranus");  
        System.out.println("8) Neptune");  
  
        Scanner userInputObj = new Scanner(System.in);  
        System.out.println("Enter the name of the planet: ");  
        String planet = userInputObj.nextLine();  
        planetFile = planet.toLowerCase() + ".txt";  
  
        factFile(planetFile);  
  
    }  
  
}
```



Viimein, kutsu aliohjelmaa pääohjelmassa, main method().

Muutoinohjelma ei aja aliohjelmaa, etkä näe tietoja ruudulla.

Tähän mennessä...

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

    while (true) {

        String planetFile;

        System.out.println("\n\n --- PLANET FACT FILE ---");
        System.out.println("1) Mercury");
        System.out.println("2) Venus");
        System.out.println("3) Earth");
        System.out.println("4) Mars");
        System.out.println("5) Jupiter");
        System.out.println("6) Saturn");
        System.out.println("7) Uranus");
        System.out.println("8) Neptune");

        Scanner userInputObj = new Scanner(System.in);
        System.out.println("Enter the name of the planet: ");
        String planet = userInputObj.nextLine();
        planetFile = planet.toLowerCase() + ".txt";

        factFile(planetFile);

    }

}
```

```
public static void factFile(String planetFile) {

    Map<String, String> dictionary = new HashMap<String, String>();

    try {
        File fileObj = new File(planetFile);
        Scanner scannerObj = new Scanner(fileObj);

        while (scannerObj.hasNextLine()) {
            String data = scannerObj.nextLine();
            //System.out.println(data);
            String[] templine = data.split(": ");
            dictionary.put(templine[0], templine[1]);
        }
        scannerObj.close();
        String planetName = dictionary.get("planetName");
        String radius = dictionary.get("radius");
        String orbitalPeriod = dictionary.get("orbitalPeriod");
        String distanceToSun = dictionary.get("distanceToSun");
        String surfaceArea = dictionary.get("surfaceArea");
        String lengthOfDay = dictionary.get("lengthOfDay");
        String density = dictionary.get("density");
        String description = dictionary.get("description");
        String funFact = dictionary.get("funFact");

        // Display Fact File

        System.out.println("\n --- " + planetName + " --- \n");
        System.out.println("# " + description + "\n");

        System.out.println(" --- How far from the sun? ---");
        System.out.println("# " + planetName + " is " + distanceToSun + " from the sun.\n");

        System.out.println(" --- Boring Maths Info --- ");
        System.out.println("# It has a surface area of " + surfaceArea);
        System.out.println("# It has a radius of " + radius);
        System.out.println("# It has a density of " + density);

        System.out.println("\n --- Extra Info ---");
        System.out.println("# The days are " + lengthOfDay + " long. 1 year is the same as " + orbitalPeriod);
        System.out.println("# " + funFact);

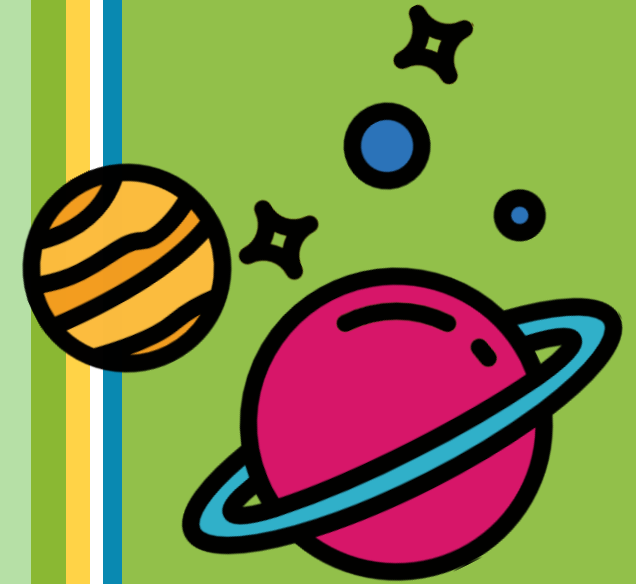
        Scanner pressEnter = new Scanner(System.in);
        System.out.println("\n-- Press enter to continue --");
        pressEnter.nextLine();

    } catch (FileNotFoundException e) {
        System.out.println("An error occurred");
    }

}
```

Lopuksi

- Mitä osaat
 - ✓ Osaat tehdä silmukan Java-kielellä
 - ✓ Osaat lukea ulkoisia tekstitiedostoja
 - ✓ Osaat soveltaa ohjelmointitaitoja oikean elämään ongelmiin.
 - ✓ Osaat tehdä vuorovaikutteisen valikon



Arkipäivän elämässä...

Muiden auttaminen



Avaruustutkimus
olisi mahdotonta
ilman
ohjelmointia!

Opettaminen muille
avaruudesta auttaa
kouluttamaan seuraavan
sukupolven
astronautteja ja
tieteentekijöitä.

Onnittelut!

Olet tehnyt planeettatietokannan!

