Almuthanna Jamal Aljajan

Oblig 3

Teori

Oppgave 1.1 - Ord og begreper

Lag deg en oversikt over hva følgende ord/begreper betyr:

- **Refaktorere:**

  er prosessen med å restrukturere eksisterende kode uten å endre den ytre atferden. Refactoring er ment å forbedre kodedesignen, gjøre det enkelt å forstå og utvide ved å fjerne den dupliserte koden og lange metoder. Fordelene inkluderer forbedret kodelesbarhet og redusert kompleksitet; disse kan forbedre vedlikehold av kildekoder og skape en mer uttrykksfull intern arkitektur eller objektmodell for å forbedre utvidbarheten.

- **Static (variabel, metode)**

  Static er et nøkkelord i Java som kan bruke både med variabler og metoder. Static variabler og metoder hører til en klasse, så de ikke tilhører en enkelt instance av en klasse. For å kalle en statisk metode eller bruke en statisk variabel, brukes klassens navn i stedet for instance navnet, Eksempel:

  Math.PI;

  Math.sqrt ();

  Static variabler kalles også klassevariabler, og vi bruker den for egenskapen som er felles for alle objekter. Static metoder kan få tilgang til static variabler og static metoder direkte, men de kan ikke få tilgang til instance metoder og instance variabler direkte. De må bruke referanse til objekt. Og static metode kan ikke bruke this nøkkelord, da det ikke er noen instance for "this" å referere til.

- **Final (variabel, metode, klasse)**

  Det final nøkkelordet i java kan brukes i mange sammenhenger med en variabel, en metode eller en klasse.

  Når en variabel er deklarert med final kan den ikke endres etter den er initialisert. Final kan brukes sammen med static og vi kaller dette ofte konstanter og navn til konstanter skrives med bare store bokstaver

Når en metode er deklarert med det final nøkkelordet. så metoden kan ikke overstyres (overridden). Når en klasse er deklarert med det final nøkkelordet, så klassen ikke kan extends (arves).

- **Abstract (klasse, metode)**

  En abstrakt klasse er en klasse som er deklarert med det abstract nøkkelordet. det kan ha et datamedlem, abstrakt metode, ikke-abstrakt metode og / eller konstruktør. Abstrakte klasser kan ikke bli instansieres, men de kan extends (arv). Abstrakt nøkkelord brukes også til å erklære en metode som abstrakt.en abstract metode inneholder bare metode signatur (hode), men ingen metodekropp, slik at en abstrakt metode vil ha en semikolon på slutten i stedet for klammeparentes. når en sub-klasse arver en abstrakt klasse, så sub-klassen må gi implementeringer til de abstrakte metodene i den abstrakte klassen hvis den har det.

- **Interface**

  Interface deklareres ved å bruke interface nøkkelordet. Det gir konstanter og abstrakte metoder. alle metodene i et interface er deklarert med bare meyode signatur uten kroppen til metoden, og alle variablene er public, static og final som standard. Et Java-interface kan implementeres ved hjelp av nøkkelord “implements ”. En klasse som implementerer etinterface, må implementere alle metodene som er deklerert interface. Java-klassen kan implementere flere Java-interface på samme tid.

  Oppgave 1.2 - Sammenligning

  getSmallestPlanet()

  Jeg har definert den metode getSmallestPlanet() som det vil ta en Arraylist av planet som en parameter, og det vil returnere et objekt av typen Planet som er den minste i ArrayList. For å oppnå det, har jeg deklarert en referansevariabel av typen Planet, og jeg gir den en initial verdi som er det første objektet i ArrayListen. Så brukte en vanlig løkke for å irritere over Arraylisten men jeg lar det å starte irritere fra det andre elementet i ArratListen. Grunnen til det fordi jeg har gitt verdien av det første elementet til det objektet som jeg har deklarert utenfor for-loopen. I hver runde vil den ta et objekt fra arrayList og sammenligne det med objektet jeg har deklarert basert på radius, og vi vil møte tre situsjoner.

  - Den første situasjonen, hvis objektet i listen er større enn objektet jeg har deklarert, så for-loop bare vil flytte til et annet element i ArrayList for å sammenligne det.
  - Den andre situasjonen er at hvis objektet i listen er miste enn objektet jeg har deklarert, vil det gi objektet jeg har deklarert den nye verdien av minste objektet og for-loop vil flytte til et annet element i ArrayList for å sammenligne den.

- den tredje situasjonen er hvis objektet i listen er lik objektet jeg har deklarert med radius, så det vil sammenligne dem med masse og sjekke hvilken av dem som har den minste massen og resign objektet jeg har deklert til objektet som har minste massen. på slutten vil denne metoden returnere den minste planeten i Array-listen.

```
public Planet minstePlanetenIsolarSystem(ArrayList<Planet> planets) {
    Planet minstePlaneten = planets.get(0);
    for (int i = 1; i < planets.size(); i++) {
        if (planets.get(i).getRadius() < minstePlaneten.getRadius()) {
            minstePlaneten = planets.get(i);
        } else if (planets.get(i).getRadius() == minstePlaneten.getRadius()) {
                if (planets.get(i).getMass() < minstePlaneten.getMass()) {
                    minstePlaneten = planets.get(i);
                }
        }
    }
    return minstePlaneten;
}
```

getSurfaceGravity()

Jeg har definert den metode getSurfaceGravity() det vil returnere en double verdi. For å finne surface gravityfor et planet, har jeg deklarert en variabel av typen double som holder gravitasjonkonstanten verdi (0.00000000006674). deretter definerte en annen variabel som skal multiplisre gravitasjonKonstanten med massen til objektet som kaller denne metoden og dividere deretter dette med Math.pow som skal to parametre. det første er radiusen som skal multipliserer med 1000 for å konertere dette fra Km til m til og den andre er nummer 2. så skal denne metode retunere verdien til den andre variabel som er surface gravity for dette planetet.

```
public double surfaceGravity() {
    double gravitasjonKonstanten = 0.00000000006674;
    double surfaceGr = ( gravitasjonKonstanten * this.getMass() ) / ( Math.pow((
this.getRadius() * 1000 ), 2) );
    return surfaceGr;
}
```

getMassInMSun()

Jeg har definert den metode getMassInMsun() at det vil returnere en double verdi. For å finne massen til et star på enheten Solar Mass (Msun), har jeg definerte en variabel inni metoden som skal dele massen til det objektet som kaller denne metoden med massen til sola. deretter skal metoden retunere verdien til den variabelen.

```
public double msunStar() {
    double mSun = this.getMass() / 1.98892E30;
    return  mSun;
}
```

Jeg har gått sammen med Sadaq Dhiblawe. forskjellen mellom koden min og hans at han brukte:

I metoden getSurfaceGravity (), foreach loop, men jeg har brukt vanlig for-loop. Den andre forskjellen er at jeg har lagret verdien av gravitasjon konstanten i en variabel, men det er han ikke. Jeg synes om par-programmering er en god ide, og det er nyttig fordi vi kan dele kunnskapen og erfaringen mellom oss og programmere med mer effekt måte.

Oppgave 1.3 – Klassediagram