Uke 3

Lite tips om Java syntaks

- Søk i W3schools om syntaks man lurer på
- https://www.w3schools.com/java/

Upcasting og downcasting med pekere

- Se på forskjellige klasser med forskjellige “briller” via peker
  - Peker har nemlig også en type, ikke bare objektet i seg selv
- Snevrer mann inn, må det skje eksplisitt
- Går det oppover, skjer det implisitt (dvs. automatisk)
- instanceof() for å sjekke hvilken type objekt det er

Private, protected og public modifikatorer

- Private
  - Bare tilgjengelig inni klassen
- Protected
  - Alle subklasser og det som er i samme “pakke” har det tilgjengelig
- Public
  - Tilgjengelig for alle

Konstruktøren i subklasser

- Bruk super() i konstruktøren for å kalle superklassens konstruktør
-

Abstrakte klasser, Object og final nøkkelord

- Abstrakte klasser har man ikke lov å lage objekt av
- Object er det alle klasser arver fra
- Final for konstant, noe som kan hjelpe med oversikt (man har kontroll på hva som er mutable og ikke mutable)

Oppgave 1

Lag et program som administrerer en samling (HashMap) av kjøretøy. Programmet skal ha følgende klasser:

1. Kjøretøy-klasse:
    - Egenskaper: id (String), merke (String), modell (String), år (int)
    - Konstruktør: for å initialisere egenskapene
    - Hent-metoder og sett-metoder for egenskapene (kun hent-metode for id)
2. Bil-klasse (arver fra Kjøretøy):
    - Ekstra egenskap: seter (int)
    - Konstruktør: for å initialisere egenskapene (inkludert de arvede egenskapene fra Kjøretøy)
    - Hent-metode og sett-metode for egenskapen seter
3. Motorsykkel-klasse (arver fra Kjøretøy):
    - Ekstra egenskap: motortype (String)
    - Konstruktør: for å initialisere egenskapene (inkludert de arvede egenskapene fra Kjøretøy)
    - Hent-metoder og sett-metoder for egenskapene
4. Program-klasse:
    - Main-metode: Opprett en samling (HashMap) for å lagre kjøretøy. Legg inn to instanser (objekter) av Bil og to instanser (objekter) av Motorsykkel. Skriv ut detaljene for kjøretøyene i samlingen.

Oppgave 2

Du lever i en verden der kunstig intelligens florerer. Enkelte roboter er mer høyteknologiske enn andre. Her skal vi ta for oss GPT, men merk at det kan eksistere roboter med navn som starter med noe annet enn GPT. Ved opprettelse av objekter, tar konstruktøren til hver robot inn et navn. Samtidig skal det ikke være mulig å opprette instanser/objekter av Robot. Det skal også finnes en metode som returnerer navnet til roboten.

GPT 3, GPT 3.5 og GPT 4 har alle noen delte egenskaper. For eksempel kommuniserer de med mennesker. Altså har de et navn og en samling *delte* egenskaper. I tillegg har GPT 3, GPT 3.5 og GPT 4 en samling *unike* egenskaper de ikke deler med de andre. Når vi snakker om samling, mener vi her enten array eller liste.

a. Tegn et klassehierarki over robotene i robotfamilien beskrevet ovenfor.
b. Implementér klassene med utgangspunkt i klassehierarkiet og informasjonen du har
c. Opprett metoder for å legge til og fjerne egenskaper fra de ulike robotene. Du kan anta at egenskaper som legges til ikke finnes fra før, og du kan anta at egenskaper som skal fjernes eksisterer.
d. Skriv og utfør enhetstester, som tester klassene og metodene du har implementert. Hvorfor er enhetstesting nyttig? Diskutér gjerne med en medstudent.