Seminaroppgaver, uke 4 - Arv og polymorfi

1. **Arv og subklasser | Egenrefleksjon og plenumsdiskusjon**
   Bruk ett minutt på å skrive ned alt du husker om arv og subklasser. Ikke stopp - du må skrive hele minuttet. Deretter skal dere diskutere det dere husker i plenum.

2. **Instanceof og typekonvertering | Diskusjon**
   Tips: Tegn et klassehierarki

a) Diskuter utsagnene i følgende kode:

```
class A {}
class B extends A {}
class C extends A {}
class D extends C {}

class TrueOrFalse {
        public static void main(String[] args) {
                A a = new A();
                B b = new B();
                C c = new C();
                D d = new D();
                A e = new D();
                A f = new B();
                System.out.println("a) " + (b instanceof A));
                System.out.println("b) " + (c instanceof D));
                System.out.println("c) " + (d instanceof A));
                System.out.println("d) " + (e instanceof C));
                System.out.println("e) " + (f instanceof C));
        }
}
```

b) I hvert av tilfellene under, diskuter om koden kommer eller ikke kommer til å krasje. Forklar hvorfor krasj oppstår eller ikke. Test i Java etter dere har diskutert alle tilfellene.

```
//Tilfelle 1
B ab = (B) a;

//Tilfelle 2
if( e instanceof B ){
   B eb = (B) e;
}

//Tilfelle 3
D ed = (D) e;

//Tilfelle 4
if( e instanceof D ){
   D ed = (D) e;
}

//Tilfelle 5
B fb = (B) f;

//Tilfelle 6
C ec = (C) e;
```

b) Vil dere si at tilfelle 3 eller tilfelle 4 er den beste løsningen? Begrunn svaret.

3. Parkeringshus med motorsykler og biler | Parprogrammering

Dere skal nå lage et program som visualiserer et parkeringshus. Et parkeringshus har parkeringsplasser hvor både Motorsykler og Biler kan parkere. Motorsykler og Biler har en del til felles, bl.a. har de begge en sjåfør og et registreringsnummer. Mens en bil også skal kunne holde på alt fra 2 til 5 personer (spesifisert i det bilen lages), derav en av de må være sjåføren.

a) Tegn klassehierarkiet

b) Lag alle klassene med instansvariabler. Inntil videre skal de ikke ha noen metoder. Det kan være lurt at Motorsykkel og Bil arver av samme klasse - hvorfor tror du det?

c) Parkeringshus skal ha to metoder: en for å sette inn en bil eller motorsykkel og en for å fjerne en bil eller motorsykkel på en gitt plass. Vi vil at metoden skal returnere bilen/motorsykkelen som står på plassen, og returnere null hvis det hverken finnes bil/motorsykkel på plassen. Når dere skal sette inn en bil eller motorsykkel, skal det gjøres på en ønsket plass. Lag unntakshåndtering til parkeringshuset ditt, for å håndtere tilfeller der man prøver å sette inn når det er fullt, eller på en ugyldig plass. **Hint:** ArrayIndexOutOfBoundsException.

d) Lag en metode skrivInformasjon(), som skriver ut all informasjon om hvert av kjøretøyene. Husk at motorsykler og biler har noe felles informasjon, siden begge er kjøretøy, men de har også noe særegen tilleggsinformasjon. Når du itererer gjennom kjøretøyene, skal all informasjon om hvert enkelt kjøretøy skrives til terminalen. Du kan ikke bruke instanceof i denne oppgaven, og du skal unngå å repetere kode unødig.

4. Egen oppgave om polymorfi | Oppgavelaging, løsning og sammenligning

Lag en egen oppgave der du bruker polymorfi. Du står fritt til å velge tema selv, men her har du noen tips: iPhone og forskjellige iPhone-modeller, kafé og ulike kafétyper, teknologibedrift med ulike avdelinger, diverse klesplagg. Etter å ha laget oppgaven, løs den selv og be noen andre om å løse den. Sammenlign deretter de to løsningene deres. Hva har dere lært av å se hverandres løsninger? Hvilke endringer ville du implementert i din egen løsning, og hvorfor? Oppgaven trenger ikke å være avansert.