

Opgave i Objektorienteret Programmering

1. semester RobTek-diplom

Denne opgave omhandler relationer mellem klasser og objekter i Java: arv, aggregering og association.

Der ønskes skrevet en applikation, som kan håndtere en kunstværkssamling.

Ideen er, at man skal bruge applikationen til at håndtere tre forskellige typer af kunstværker: bøger, kompositioner og malerier. Disse er i applikationen implementeret som subklasser til den abstrakte superklasse *Kunstvaerk*.

Denne klasse indeholder som minimum attributterne *titel* (String), *aar* (int) og *ophavsmænd*, som er et array af klassen *Person*, der kan indeholde op til fem elementer¹.

Klassen *Bog* indeholder attributterne *genre* (String) og *antalSider* (int).

Klassen *Komposition* indeholder attributterne *genre* (String) og *spilletid* (int).

Klassen *Maleri* indeholder attributterne *laengde* (int), *bredde* (int) og en reference (association) til et objekt af klassen *Museum*, som indeholder følgende oplysninger om, hvor maleriet er udstillet: *navn* (String) og *adresse* (String).

Den sidste klasse, *KunstvaerkSamling*, indeholder en enkelt attribut *Kunstvaerker*, som er et ArrayList objekt eller et array, som består af objekter af klassen *Kunstvaerk*.

1.

Alle klasserne skal skrives, således at nedenstående *test-driver* kan afvikles korrekt.

```
public class TestDriver
{
    public static void main(String args[])
    {
        Museum smk = new Museum ("Statens Museum for Kunst", "Sølvtorvet,
                                   København");
        Museum fsm = new Museum ("Fyns Stifts Museum", "Jernbanegade, Odense");

        Person lar = new Person ("L.A. Ring", "dansk", 1854, 1933);
        Maleri kal = new Maleri ("Kalkemanden", 1908, 80, 65, smk);
        kal.addOphavsmand(lar);

        Person cwe = new Person ("C. W. Eckersberg", "dansk", 1783, 1853);
        Maleri rus = new Maleri ("Et russisk linjeskib til ankers ved
                                   Helsingør", 1828, 140, 135, fsm);
        rus.addOphavsmand(cwe);

        Person køb = new Person ("Christian Købke", "dansk", 1810, 1848);
        Maleri par = new Maleri ("Parti fra Dosseringen mod Nørrebro",
                                   1841, 100, 85, smk);
        par.addOphavsmand(køb);

        KunstvaerkSamling kvs = new KunstvaerkSamling();
        kvs.add(kal);
        kvs.add(rus);
        kvs.add(par);
    }
}
```

¹ Der skal kunne tages højde for det forhold, at fx en komposition har mere end én komponist

```

Person jl = new Person ("John Lennon","britisk",1940,1980);
Person pm = new Person ("Paul McCartney","britisk",1942,0);
Komposition y = new Komposition("Yesterday",1965,"Rock",211);
y.addOphavsmand(jl);
y.addOphavsmand(pm);

Person db = new Person ("David Bowie","britisk",1947,2016);
Komposition lom = new Komposition("Life on Mars",1971,"Rock",309);
lom.addOphavsmand(db);

kvs.add(y);
kvs.add(lom);

Person mah = new Person ("Martin A. Hansen","dansk",1909,1955);
Bog l = new Bog("Løgneren",1950,"Prosa",232);
l.addOphavsmand(mah);

Person hs = new Person ("Hans Scherfig","dansk",1905,1978);
Bog dff = new Bog("Det forsømte Forår",1940,"Prosa",218);
dff.addOphavsmand(hs);

kvs.add(l);
kvs.add(dff);

}
}

```

2

I klassen *KunstvaerkSamling* skal der programmeres tre metoder, som alle skriver deres resultater på *System.out*. For at løse opgaven, skal der muligvis laves flere metoder end dem, der er brugt i test-driveren.

a

```
public void findSangeMedFlereKomponister();
```

På ovenstående testdata udskriver metoden navnet på sangen "Yesterday"

b

```
public void findMalerierPaaMuseum(String);
```

På ovenstående testdata udskriver metoden kaldt med parameteren "Statens Museum for Kunst" navnene "Kalkemanden" og "Parti fra Dosseringen mod Nørrebro"

c

```
public void beregnAntallæsteSider();
```

På ovenstående testdata udskrives tallet 450, som er summen af sider i bøgerne.