

# Inlämningsuppgift 5

Datateknik GR(A), Java I, 7,5 högskolepoäng

Syfte:	Att lära sig använda arv för att utöka befintliga klasser. Att använda verktyget jar för att skapa jar-filer.
Att läsa:	Lektion 5
Uppgifter:	1
Inlämning:	Inlämningslåda 5 i Moodle

Lycka till!



## Uppgift 1

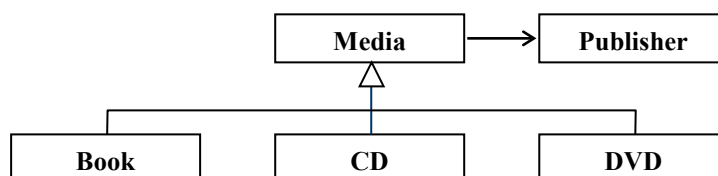
### Översikt

Denna uppgift går ut på att skapa en arvshierarki bestående av en superklass samt tre subclasser. Syftet med laborationen är att förstå och kunna använda begreppet arv.

### Uppgift

I inlämningsuppgift 3 har du skapat klasser för att hantera CD-skivor och de förlag dessa är utgivna på. I denna uppgift ska du bygga vidare på detta. Du ska utöka klasserna så att det även går att hantera böcker och DVD-skivor. Detta ska göras med hjälp av arv. En generell superklass, `Media`, ska skapas för att kunna återanvända gemensamma data och funktionalitet mellan klasserna `CD`, `Book` och `DVD`. Dessa tre klasser ska alltså ärva sina egenskaper från klassen `Media`.

Du ska skapa en arvshierarki som ser ut enligt nedan:



Klassen `Media` är en superklass för olika typer av media. Här ska du samla instansvariabler och metoder som är gemensamma för alla typer av media. Som du ser så kan du återanvända en hel del från klassen `CD` som du skrev i inlämningsuppgift 3.

Alla instansvariabler i klassen är deklarerad som `protected` (markeras med `#` i UML), vilket innebär att klasser som ärver från `Media` har en direkt tillgång till dessa variabler.

Klassen behöver endast två konstruktorer. Den ena tar tre argument som sätter värden på klassens alla instansvariabler, och den andra tar inga argument utan sätter vissa standardvärden på instansvariablerna. Anser du att fler konstruktorer är nödvändiga får du givetvis lägga till dessa.

Media
<pre># title : String # length : int # publisher : Publisher</pre>
<pre>+ Media() + Media(title : String, length : int,         publisher : Publisher) + setTitle(title : String) + setLength(length : int) + setPublisher(publisher : Publisher) + setPublisher(name : String, vphone : String) + getTitle() : String + getLength() : int + getPublisher() : Publisher + print() + toString() : String</pre>

I metoden `setLength` ska du utföra en kontroll så att det inskickade värdet inte är mindre än noll. Det ska inte heller gå att skapa ett `Media`-objekt vars längd är mindre än noll. Det finns ingen mening att sätta längden på ett media till `-1`. Om värdet som skickas till metoden är mindre än noll ska värdet sättas till `0` (noll). Om värdet som skickas till metoden är större än noll kan vi tilldela instansvariabeln detta värde.

Första subclassen du ska skapa är `DVD`. Denna klass innehåller endast en "egen" instansvariabel, `ageLimit`, som används för att ange åldersgränsen på en video. Det

ska även finnas `set-` och `get-`metoder för denna instansvariabel. I övrigt ärvs alla egenskaper från klassen `Media`. Längden på en dvd lagras i klassen som totala antalet sekunder.

I metoden `setAgeLimit` ska du göra en kontroll så att det inskickade värdet inte är mindre än noll, d.v.s. samma kontroll som utförs för metoden `setLength` i klassen `Media`.

DVD
- ageLimit : int
+ DVD() + DVD(title : String, length : int, ageLimit : int, publisher : Publisher) + setAgeLimit(ageLimit : int) + getAgeLimit() : int + getLengthAsString() : String + print() + toString() : String

Vissa metoder behöver du överskugga i `DVD` för att få det beteende vi önskar. Till att börja med ska metoden `toString` returnera en lämplig strängrepresentation av aktuellt dvd-objekt.

Metoden `print` ska skriva ut objektets alla värden samt anropa metoden `print` i superklassen för att skriva ut dess värden.

Du måste även skriva en metod `getLengthAsString()`. Metoden ska returnera längden på dvd-skivan som en sträng enligt formatet: **HH**<sub>h</sub> **MM**<sub>m</sub> **SS**<sub>s</sub>.

Strängen med timmar (HH), minuter (MM) och sekunder (SS) ska alltid minst innehålla två siffror. D.v.s. i de fall timmarna, minuterarna eller sekunderna är mindre än 10 ska det sättas ut en nolla framför. Exempel på strängar som kan returneras är följande:

```
00h 54m 09s    // (När length = 3249)
01h 09m 45s    // (När length = 4185)
100h 00m 00s   // (När length = 360000)
```

Det ska finnas minst två olika sätt att skapa objekt av `DVD`. En konstruktor som tar fyra argument och en som inte tar några argument alls. Sen är det upp till dig om du önskar fler sätt att skapa objekt på. Det ska inte vara möjligt att skapa en DVD vars längd eller åldersgräns har ett negativt värde (ändra i så fall värdet till 0).

Den andra subklassen du ska skapa är `Book` vilken innehåller en instansvariabel, `author` som är författarens namn. Precis som för `DVD` kommer denna klass att ärva sina egenskaper från `Media`, men att `set-` och `get-`metoder för `author` ska finnas.

Book
- author : String
+ Book() + Book(author : String, title : String, length : int, publisher : Publisher) + setAuthor(author : String) + getAuthor() : String + getLengthAsString() : String + print() + toString() : String

Metoderna `toString` och `print` ska överskuggas på samma sätt som i `DVD` och `getLengthAsString` ska returnera bokens längd som en sträng enligt följande format:  
`xxx page(s)`

Det vill säga, om antalet sidor endast är 1 skrivs `page` ut efter sidantalet (alternativt att sida skrivs ut om du har svenska utskrifter i övrigt). Är antalet sidor fler än 1 skrivs `pages` ut efteråt

(alternativt sidor).

Exempel på strängar som kan returneras är:

```
0 pages    // (length = 0)
1 page     // (length = 1)
321 pages  // (length = 321)
```

Konstruktörer för att skapa objekt av `Book` ska finnas där ena har fyra parametrar och den andra inte har några parametrar alls. Önskar du fler sätt att skapa böcker på är det fritt att lägga till konstruktörer för detta.

Den tredje och sista subklassen är `CD` som du inte behöver göra några större förändringar i. Tack vare superklassen `Media` kan du ta bort vissa instansvariabler och metoder från din ursprungliga `CD`-klass.

En del metoder behöver du överskugga så de passar in i arvsstrukturen. Dessa är, precis som för `DVD` och `Book`, metoderna `print` och `toString`.

CD
- artist : String
+ CD() + CD(artist : String, title : String, length : int, publisher : Publisher) + setArtist(artist : String) + getArtist() : String + getLengthAsString() : String + print() + toString() : String

Implementation av `getLengthAsString` ska returnera längden på `CD:n` som en sträng enligt formatet: **HH:MM:SS**.

Strängen med timmar (HH), minuter (MM) och sekunder (SS) ska alltid innehålla minst två siffror. D.v.s. i de fall timmarna, minuterarna eller sekunderna är mindre än 10 ska det sättas ut en nolla framför. Exempel på strängar som kan returneras är följande:

```
00:54:09    (När length = 3249)
01:09:45    (När length = 4185)
100:00:00   (När length = 360000)
```

Som vanligt ska det finnas minst två olika sätt att skapa objekt av `CD`. En konstruktor som har fyra parametrar och en som inte har några alls. Sen är det upp till dig om du önskar fler sätt att skapa objekt på.

För att kontrollera att allting fungerar som tänkt ska du skriva en testklass som skapar minst två objekt vardera av `Book`, `DVD` och `CD` (d.v.s. totalt minst sex objekt). Kalla testklassen för t.ex. `MediaTest`. Denna testklass ska fungera på samma sätt som i inlämningsuppgift 3 (`CDTest`). D.v.s. den ska på lämpligt testa alla metoder som finns i klasserna du skrivit till denna inlämningsuppgift. Inget krav finns att använda inmatning för att ge objekten värden.

När allting är klart ska du skapa en jar-fil som innehåller alla .class-filer som ingår i uppgiften. jar-filen ska innehålla en manifest-fil som anger din testklass som `Main-Class`.

## Krav

För att uppgiften skall betraktas som godkänd krävs ovan nämnda detaljer samt detta:

- `protected` ska användas för instansvariabler i klassen `Media`
- Inga publika instansvariabler får i övrigt finnas i någon klass, endast metoder får vara publika.
- Klasser och metoder behöver inte kommenteras med dokumentationskommentarer om du inte vill.
- I subklassernas `print` metod ska längden på aktuellt media skrivas ut med den sträng som metoden `getLengthAsString` returnerar. För `CD` och `DVD` ska utskriften ske med två siffror för timmar, minuter och sekunder (exempel 00, 02, 10, 23 etc).
- I subklassernas `print`-metod ska ett anrop till superklassens `print` ske.
- I subklassens konstruktor ska en lämplig konstruktor i superklassen anropas.
- Alla skärmutskrifter skall vara välstrukturerade och lätta att förstå.
- För metoderna `setLength`, och `setAgeLimit()` ska det inte vara möjligt att sätta ett negativt tal.

## Tips

- Använd modulus (%) tillsammans med division för att dela upp längden (antal sekunder) till timmar, minuter och sekunder.
- Eller prova vad följande programsatser gör:

```
String.format("%02d", 9);
```

- Uppmana gärna användaren att trycka på t.ex. **enter** vid jämna mellanrum för att fortsätta utskriften i `MediaTest` (så inte utskrifterna blir allt för långa).