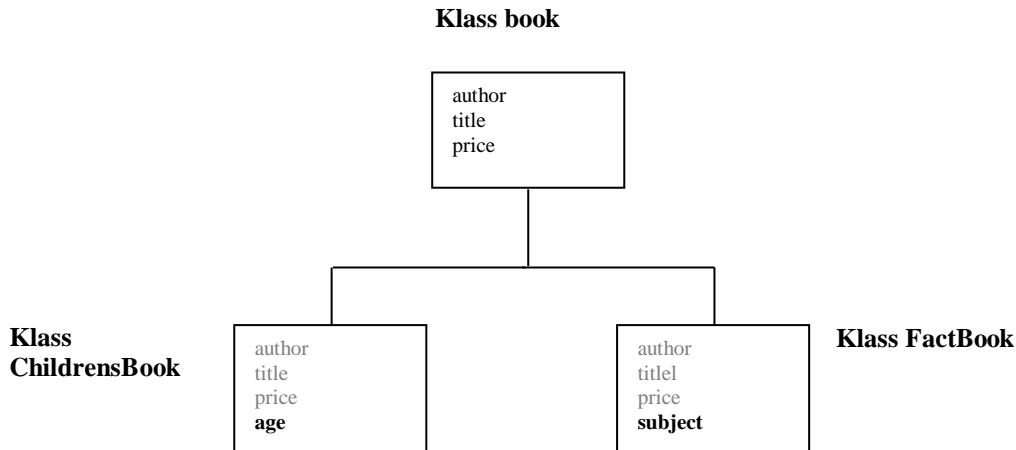


Objektorientering del 3 Java – Arv

När man gör en ny klass behöver man inte alltid börja från början. Genom arv kan man ärva funktionalitet som finns i en existerande klass.

Ibland säger man föräldraklass eller basklass och barnklass.

Ex.



Eftersom barnbok och faktabok ärver från bok så har de samma funktionalitet. Man kan säga att en barnbok eller faktabok också **är** en bok.

Man kan sedan lägga till funktionalitet i de ärvda klasserna, som ålder för barnbok eller ämne för faktabok.

För att ärva skriver man skriver man extends. Extends betyder utöka och det är det vi gör. Vi utökar basklassen eller som det kallas i Java superklassen.

Så här: `class ChildrensBook extends book`

Här är klassen som vi ska ärva från:

```
class Bok
{

    public String ISBN;
    public String titel = "N N";
    public String author = "N N";

    public Bok()
    {
    }

    public Bok(String ISBNnr, String titeln, String författare)
    {
        setISBN(ISBNnr);
        setTitel(titeln);
        setAuthor(författare);
    }

    // Setters

    public void setISBN(String inISBN)
    {
        ISBN=inISBN;
    }
    public void setTitel(String inTitel)
    {
        titel=inTitel;
    }
    public void setAuthor(String inAuthor)
    {
        author=inAuthor;
    }

    public void setKund(Deltagare inKund)
    {
        Bibliotekskund = inKund;
    }

    // Getters

    public String ISBN()
    {
        return ISBN;
    }
    public String Titel()
    {
        return titel;
    }
    public String Author()
    {
        return author;
    }

}
```

Här

```
class ChildrensBook extends Book
{
    private string age;

    public string Age
    {
        get
        {
            return age;
        }
        set
        {
            age = value;
        }
    }
}

//Konstruktor (constructor) Metod som körs när objektet skapas. Denna är tom.
```

```

public ChildrensBook ()
{
}

public ChildrensBook (string titel, string author, int price, string age)
{
    Author = author;
    Titel = titel;
    Price = price;
    Age = age;
}

public override void skrivUt()
{
    base.skrivUt();
    Console.WriteLine("Ålder: {0}", Age);
    Console.WriteLine("");
}
}
}

```

För att skapa ett objekt av vår nya klass BarnBok skriver vi:

```

ChildrensBook enBarnBok = new ChildrensBook();

```

Vi ser att vi inte har deklarerat en författare (author) som vi gjorde i book:

```

private string author;

```

Trots det kan vi bestämma titeln på barnboken så här:

```

enBarnBok.Author = "Astrid Lindgren";

```

Hur kan vi göra det?

Jo, för vi har ärvt det ifrån vår basklass book.

Allt vad vår basklass kan göra kan klassen som ärver också göra.

Vitual och Override

Har man en metod som vår **skrivUt()** och vill använda den för att skriva ut allt som tillhör en bok men även det nya i barnklassen måste man använda två nyckelord: **virtual** och **override**.

Virtual skriver man i metoden som man har i basklassen och **override** skriver man i den metod man har i barnklassen.

Så här:

Basklass:

```

public virtual void skrivUt();
{
    skrivUt(); //Original-metod
}

```

```
}
```

Barnklass:

```
public override void skrivUt()
{
    base.skrivUt(); //Detta anropar skrivUt() i basklass och skriver ut det som alla böcker har gemensamt.

    //Vi kan nu lägga till det som är unikt för barnklassen:

    Console.WriteLine("Ämne: {0}", Subject);
}
```

Uppgifter

1. Öppna uppgiften från metoder och konstruktörer och skapa en klass som ärver från bok. Ni ska ha samma fältvariabler, samma egenskaper och samma konstruktörer som i klassen bok **men ingen metod skrivUt()**. Skapa sedan ett objekt med konstruktorn som tar tre inparametrar med den nya klassen.

Anropa metoden skrivUt() med objektet (trots att vi inte skrev den). Vad händer och varför?

2. När ni sätter värdet i efterhand skriver ni t.ex. HistoryBook.Titel. Vilken del av er klass är det som ni anropar då?

Titel med litet t är ju satt till private. Vad händer om ni istället försöker anropa variabeln direkt så här:
HistoryBook.titel? (Med litet t)

Varför?

3. Skapa en ny consoleapplication och döp den till Arv. Skapa en klass **Bok** och tre klasser som ärver från klassen Bok. Klassen bok ska ha fälten titel (string), författare (string), och pris (int) som tidigare. Den första barnklassen ska heta **barnbok** och extra information är **ålder** (age) som är en **int**. Den andra ska heta **FaktaBok** och extra information ska vara **ämne** (subject) som är en **string**. Den tredje ska heta **KokBok** och extra information ska vara område (area) som ska vara en **string**. Område kan vara för vegetarianer eller något annat.

Varje klass ska ha privata fält och egenskaper som sätter och hämtar värdet. Bokklassen ska ha en konstruktor som tar tre parametrar och de övriga en konstruktor som tar fyra parametrar.

Alla klasser ska ha en metod som heter **skrivUt**. Barnklasserna ska skriva ut den information som finns i basklassen och dessutom sin extra information.

4. Vi antar att priset är utan moms. Skriv en metod som räknar ut priset inklusive moms och skriver ut det. Vi kan anta att priset är 25 %.