Рад са наслеђивањем

Наслеђивање је кључни појам ООП и користи се као алатка којим се избегава понављање када се дефинишу различите класе које имају неке заједничке могућности и које су јасно међусобно повезане.
То могу бити различите класе истог типа, свака са својим могућностима које се разликују (нпр класе руководиоци, радници, запослени у истој фирми).
На примеру фабрике, ако се пише апликација која описује ту фабрику класе руководиоци и радници имају неке заједничке особине али и особине које су различите.
Нпр сви имају платни број запосленог, али различите одговорности и различите радне задатке.

Наслеђивање

Наслеђивање (inheritance) у програмирању је само класификација, однос између класа.
Нпр, и коњи и китови су примери сисара; имају све атрибуте које сисари имају (удишу ваздух, доје младунце, топлокрвни су) али имају и своје посебне особине (имају копита тј пераја).
Како се може направити модел коња и кита у програмирању?
Један начин је да се направе две различите класе под називом Коњ и Кит.
Обе класе могу да остварују понашања која су јединствена за тај тип сисара, Касање (за коња) или Пливање (за кита) на посебан начин.
Али како приступити понашањима која су заједничка и за коња и за кита као што су Дисање, ДојењеМладунаца?
Исте методе са истим називима у обе класе се морају поставити што представља велико дуплирање посла, посебно ако касније треба направити моделе других типова сисара, као што су Људи или Мравојед.

Наслеђивања помаже у решавању овог проблема, па за различите типове сисара се прави класа под именом Сисар која нуди заједничке функционалности које показују различити типови сисара.
Тада се могу декларисати класе Коњ, Кит, Људи које наслеђују из класе Сисар.
Те класе онда аутоматски обухватају функционалности класе Сисар (Дисање, ДојењеМладунаца...) али сваку од тих класа можемо проширити функционалностима кје су јединствене за поједине типове сисара – методом Касање за класу Коњ или методом Пливање за класу Кит.
Ако треба променити начин на који заједничка метода као што је Дисање ради, потребно је променити само на једном месту, у класи Сисар.

Коришћење наслеђивања

Класу која наслеђује другу класу можемо декларисати:

```
class IzvedenaKlasa: OsnovnaKlasa
{
      //…
}
```

Изведена класа наслеђује основну класу, а методе из основне класе постају део изведене класе.
У програмском језику C# дозвољено је да нека класа наслеђује највише једну основну класу; није допуштено да класа наслеђује две или више класа.

Међутим, осим ако IzvedenaKlasa није декларисана као sealed (запечаћена), можемо користити исту синтаксу за извођење других класа које наслеђују класу IzvedenaKlasa:

```
class IzvedenaPotKlasa: IzvedenaKlasa
{
      //…
}
```

Пример: декларисати класу Sisar са методама Disanje и DojenjeMladunaca које су заједничке за све сисаре

```
class Sisar
{
      public void Disanje()
      {
            //...
      }
      public void DojenjeMladunaca()
      {
            //...
      }
      //...
}
```

Сада могу да се дефинишу класе за сваки посебан тип сисара додавањем више метода по потреби:

```
class Konj: Sisar
{
      //...
      public void Kasanje()
      {
            //...
      }
}
class Kit: Sisar
{
      //...
      public void Plivanje()
      {
          //...
      }
}
```

У C# се увек подразумева да је наслеђивање јавно.

Када је направљен објекат mojKonj у апликацији, може да се позива метода Kasanje, Disanje i DojenjeMladunaca:

```
Konj mojKonj = new Konj ();
mojKonj.Kasanje();
mojKonj.Disanje();
mojKonj.DojenjeMladunaca();
```

Слично, може се направити објекат mojKit али се сада позивају методе Plivanje, Disanje и DojenjeMladunaca; метода Kasanje није доступна, пошто је дефинисана само у класи Konj.
Наслеђивање се може применити само на класе а не и на структуре.
Све структуре заправо наслеђују апстрактну класу под називом System.ValueType.

Класа System.Object

Ова класа је корен класа свих класа у C#.
Све класе су посредно изведене из класе System.Object.
Последица тога је да C# компајлер неприметно преправља класу Sisar у следећи код:

```
class Sisar: System.Object
{
     //…
}
```

Све методе у класи System.Object се аутоматски прослеђују надоле у ланцу наслеђивања класа које се изводе из класе Sisar, као што су класе Konj, Kit.
То значи да све класе које се дефинишу аутоматски наслеђују све могућности класе System.Object.
Овиме се обухватају методе, као што је нпр метода ToString.
Пример: Приказати коришћење наслеђивања на класи Zaposleni и поткласи Programeri наслеђивањем вредности поља надкласе.

```
using System;
namespace ProjekatCS002
{
    public class Zaposleni
    {
        public float plata = 100000;
    }
    public class Programeri: Zaposleni
    {
        public float bonus = 10000;
    }
    class Program
    {
        public static void Main()
        {
            Programeri radnik1 = new Programeri ();
            Console.WriteLine("Plata: " + radnik1.plata);
            Console.WriteLine("Bonus: " + radnik1.bonus);
        }
```

```
    }
}
```

Задаци за самосталан рад

1. Написати програм приказује коришћење наслеђивања поља описа топлокрвности надкласе Sisari у поткласама Konj и Kit. Поља у поткласама иницијализују вредности брзине кретања објеката и описа величине објекта (ситан, велик, огроман).

2. Написати програм којим се користи наслеђивање методе Hranjenje надкласе Zivotinje у поткласи Pas. Поткласа има своју методу Lajanje. Инстанцирати објекат класе Pas и показати да наслеђивање функционише. Све методе исписују одговарајућу поруку на екрану.

3. Доказати модификацијама кода у претходна два задатка да наслеђивање не функционише уназад, тј да надкласа не наслеђује поткласу.