

Декларисање виртуелних метода

Понекад је потребно сакрити начин на који је метода остварена у основној класи.

Метода ToString из класе System.Object претвара објекат у његову представу од низа знакова.

Верзија методе ToString зна како да претвори инстанцу у низ знакова тако што садржи назив тог објекта (Sisari или Konj).

Корисно је направити своју верзију методе ToString у свакој класи која се дефинише, чиме се надјачава подразумевана примена у класи System.Object.

Верзија у класи System.Object постоји само као заштита у случају да класа не примењује или јој није потребна посебна верзија методе ToString.

Метода која је предвиђена да буде надјачана се назива виртуелна метода; надјачавање методе је механизам којим се нуди другачија примена исте методе – све методе су повезане, пошто су предвиђене за обављање истог задатка, али на начин који је посебан за сваку класу.

Сакривање методе је средство замене једне методе другом – методе обично нису повезане и могу да обављају потпуно другачије задатке.

Надјачавање је корисно, сакривање је обично грешка.

Метода се може обележити као виртуелна коришћењем службене речи virtual:

```
class Object
{
    public virtual string ToString()
    {
        //...
    }
}
```

Декларисање надјачаних метода

Ако основна класа декларише методу која је виртуелна, изведена класа може коришћењем службене речи override да декларише другу примену ове методе:

```
class Konj: Sisari
{
    public override string ToString()
    {
        //...
    }
}
```

Нова примена методе у изведеној класи може да позове првобитну примену ове методе у основној класи коришћењем службене речи base:

```
public override string ToString()
{
    string x = base.ToString();
    //...
}
```

Постоји неколико важних правила која се користе приликом декларације полиморфне методе коришћењем кључних речи virtual и override:

- Виртуелна метода не може бити приватна, а зато и надјачане методе не могу бити приватне; надјачане методе могу имати посебан облик приватности познат као заштићени приступ (protected access)
- Потписи виртуелне и надјачане методе морају бити истоветни (исти назив, број и типове параметара) као и да враћају вредност истог типа
- Може се надјачати само виртуелна метода
- Ако изведена класа не декларише методу коришћењем службене речи `override`, она не надјачава методу основне класе, већ је сакрива
- Надјачана метода је одмах виртуелна и може да се надјача у будућој изведеној класи; не може се посебно декларисати да је надјачавање методе виртуелно коришћењем кључне речи `virtual`

Виртуелне методе и полиморфизам

Коришћењем виртуелних метода може се позивати различите верзије исте методе на основу типа објекта који се динамички одређује у тренутку извршавања.

Следећи примери класа дефинишу варијације хијерархије `Sisari`:

```
using System;
namespace ProjekatCS002
{
    class Sisari
    {
        //...
        public virtual string UzmiNazivTipa()
        {
            return "Ovo je sisar";
        }
    }
    class Konj: Sisari
    {
        //...
        public override string UzmiNazivTipa()
        {
            return "Ovo je konj";
        }
    }
    class Kit: Sisari
    {
        //...
        public override string UzmiNazivTipa()
        {
            return "Ovo je kit";
        }
    }
    class Mravojed: Sisari
    {
```

```

        //...
    }
    class Program
    {
        public static void Main()
        {
            Sisari mojSisar;
            Konj mojKonj = new Konj();
            Kit mojKit = new Kit();
            Mravojed mojMravojed = new Mravojed();
            mojSisar = mojKonj;
            Console.WriteLine(mojSisar.UzmiNazivTipa()); //Konj
            mojSisar = mojKit;
            Console.WriteLine(mojSisar.UzmiNazivTipa()); //Kit
            mojSisar = mojMravojed;
            Console.WriteLine(mojSisar.UzmiNazivTipa()); //Mravojed
        }
    }
}

```

Даје:

Ovo je konj

Ovo je kit

Ovo je sisar

Види се да метода `UzmiNazivTipa` у класама `Konj` и `Kit` има службену реч `override` као и да класа `Mravojed` нема ту методу.

Очекивано је да се прикаже искључиво `Ovo je sisar` пошто сви искази позивају методу `UzmiNazivTipa` у промењивој `mojSisar`, која је `Sisari`.

Али у првом случају промењива `mojSisar` указује на објекат `Konj` (дозвољено додељивање објекта `Konj` промењивој `Sisari` пошто `Konj` наслеђује класу `Sisari`).

Пошто је тада метода `UzmiNazivTipa` дефинисан акао виртуелна, извршни модул разуме да треба да позове методу `Konj.UzmiNazivTipa`, тако да овај исказ у ствари приказује поруку `Ovo je konj`.

Трећи исказ позива методу `ispis` на објекат `Mravojed`, али класа `Mravojed` нема методу тако да се позива методу у класи `Sisari`, која враћа `Ovo je sisar`.

Овај феномен да исти исказ позива различите методе зависно од контекста у коме се налази, назива се полиморфизам (више облика).

Задаци за самосталан рад

1. Написати програм са основном класом `Zivotinje` и поткласом `Pas`. У основној класи креирати виртуелну методу `Hranjenje` а у поткласи извршити надјачавање ове методе. Проверити исправност надјачавања у главној функцији програма.

2. Написати програм којим се коришћењем службене речи `base` приступа пољу основне класе из изведене класе. Нека су класе `Zivotinja` и `Pas`, метода у изведеној класи `PrikaziBoje` за приказ вредности поља `boja` која се налази у обе класе.
3. Написати програм који приказује употребу полиморфизма над виртуелном методом `Hranjenje` у основној класи `Zivotinje`, унутар изведене класе `Pas` надјачавањем исте методе.