KATEDRA INFORMATIKY
PŘÍRODOVĚDECKÁ FAKULTA
UNIVERZITA PALACKÉHO V OLOMOUCI

Rozhraní

7. cvičení

Jiří Zacpal

KMI/ZP3CS – Základy programování 3 (C#)

Rozhraní

- rozhraním objektu se myslí to, jak je objekt viditelný zvenku
- můžeme definovat rozhraní mimo třídu -> lze jej použít pro více tříd

Definování rozhraní

- místo class se použije klíčové slovo interface
- u metod se neuvádí modifikátor (public, protected, private)
- místo těla metody se napíše středník
- příklad:

```
interface IPorovnatelny
{
    int PorovnejS(object obj);
}
```

Implementace rozhraní

- třída či struktura dědí rozhraní
- v třídě či struktuře musí být implementovány všechny metody rozhraní, přičemž platí:
  - názvy, návratové typy a parametry metod se musí shodovat
  - před jménem metody může být uveden název rozhraní (explicitní implementace rozhraní)
  - všechny metody implementující rozhraní musí být veřejně přístupné
- příklad:

```
interface ISuchozemsky
{
    int PocetNohou();
}
class Kun:ISuchozemsky
{
    ...
    public int PocetNohou()
    { return 4;}
```

Rozhraní a dědičnost

- třída může dědit jinou třídu a současně implementovat rozhraní
- nejdříve se uvede název bázové třídy a potom rozhraní
- příklad:

```
interface ISuchozemsky
{...}
class Savec
{...}
class Kun:Savec,ISuchozemsky
{...}
```

Odkazování na třídu prostřednictvím rozhraní

- funguje stejným způsobem jako u tříd
- příklad:

```
Kun mujKun=new Kun();
ISuchozemsky iMujKun=MujKun;
```

Práce s více rozhraními

- třída může implementovat neomezený počet rozhraní
- příklad:

```
class Kun, ISuchozemsky, IPasouciSe
{
    ...
}
```

Explicitně implementované rozhraní

- pro případ více rozhraní se stejnou metodou
- příklad:

```
interface ISuchozemsky
{int PocetNohou();}
interface ICesta
{int PocetNohou();}

class Kun:ISuchozemsky,ICesta
{
        int ISuchozemsky.PocetNohou()
        {return 4;}

        int ICesta.PocetNohou()
        {return 3;}
}
```

Explicitně implementované rozhraní

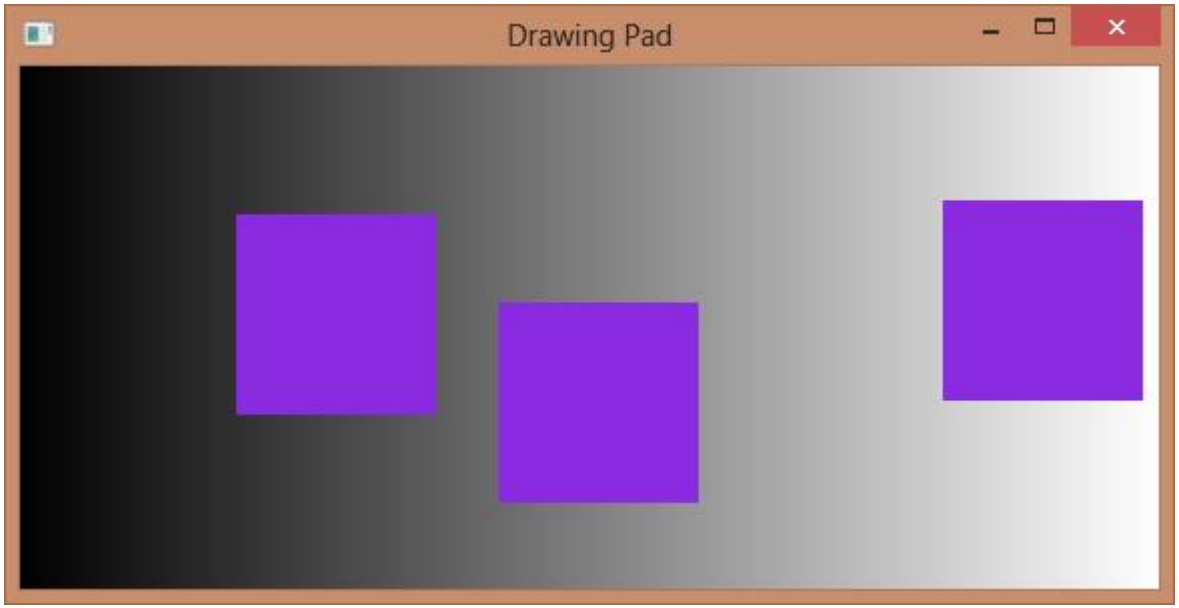
- metody nejsou veřejné a nelze je tedy volat
- přístup je možný pomocí příslušného rozhraní
- příklad:

```
Kun mujKun = new Kun();
...
ICesta cestovniKun=kun;
int nohouNaCeste=cestovniKun.PocetNohou();
ISuchozemsky suchozemskyKun=kun;
int nohouNaKoni=suchozemskyKun.PocetNohou();
```

Omezení pro rozhraní

- v rozhraní nelze definovat žádné datové položky a to ani statické
- v rozhraní nelze definovat konstruktor (ani desktruktor)
- u žádné metody nelze specifikovat přístupový modifikátor
- do rozhraní nelze vkládat žádné typy (výčtové, struktury,...)
- rozhraní nemůže dědit z třídy či struktury, může však dědit od jiného rozhraní

Příklad 1



Abstraktní třídy

- z abstraktní třídy nelze vytvářet objekty
- slouží pro společné metody tříd, které ji dědí
- příklad:

```
abstract class PasouciSeSavec:Savec,IPasoucise
{
      void IPasouciSe.Zvykej()
      {Console.WriteLine(„Žvýkám.“);}
}

class Kun: PasouciSeSavec,ISuchozemsky
{...}

class Ovce: PasouciSeSavec,ISuchozemsky
{...}
```

Abstraktní metody

- může být v abstraktní třídě
- je podobná virtuální metodě, ale neobsahuje tělo
- odvozená třída musí tuto metodu předefinovat
- příklad:

```
abstract class
PasouciSeSavec:Savec,IPasoucise
{
    abstract void StravujTravu();
}
```

Zapečetěné třídy

- nelze ji použít jako bázovou třídu
- příklad:

```
sealed class Kun:Savec,ISuchozemsky
{
    ...
}
```

Zapečetěné metody

- metoda v nezapečetěné třídě
- v odvozené třídě ji nelze předefinovat
- zapečetit lze pouze předefinovanou metodu, která se pak definuje jako selaed override