

26 Rozhranie

26.1 Rozhranie – interface.

V predchádzajúcej kapitole sme sa dozvedeli, že trieda môže vzniknúť odvodením od inej triedy, čím získa členy rodičovskej triedy, môže doplniť svoje vlastné členy a predefinovať metódy rodičovskej triedy. Tiež sme si ukázali, že konštruktory rodičovskej triedy sa nededia, ale môžu byť volané v potomkovi metódou *super()*. Spomenuli sme tiež finálne metódy (nie je možné ich v potomkovi predefinovať) a finálne triedy (nie je možné ich dediť) a takisto abstraktné metódy (musia byť vždy predefinované) a abstraktné triedy (dá sa z nich len dediť, nemôžu vytvárať inštancie).

V Jave môže trieda dediť iba z jednej triedy, hovoríme o jednoduchovej dedičnosti. Viacnásobnú dedičnosť, t.j. dedenie z viacerých tried vieme v Jave realizovať pomocou **rozhraní**.

Rozhranie sa síce syntakticky podobá triede, medzi triedou a rozhraním sú však podstatné rozdiely:

- v deklarácii rozhrania sa používa rezervované slovo **interface**
- metódy rozhrania neobsahujú žiadne telo
- v rozhraní nemožno deklarovať premenné, iba konštanty

Z uvedeného vyplýva, že rozhrania predstavujú len akési šablóny určené pre triedy, ktoré ich implementujú.

Rozhrania používame najmä v týchto dvoch prípadoch:

- keď trieda implementuje vlastnosti niekoľkých iných tried
- keď potrebujeme vytvoriť viacero tried s podobnými vlastnosťami.

26.2 Vytvorenie rozhrania. Rezervované slovo interface.

Ak chceme s rozhraním pracovať musíme ho, podobne ako triedu, najskôr vytvoriť. Postup pri vytvorení je rovnaký, ako pri vytvorení triedy, len druhom kroku volíme Java Interface:

1. v ľavej časti obrazovky, v záložke **Projects** vyvoláme na názve balíka kontextové menu
2. postupne zvolíme: **New / Java Interface...**
3. zapíšeme názov rozhrania a klikneme na **Finish**

NetBeans vygeneruje nasledovný kód:

```
...
public interface NazovRozhrania{

}
```

26 Rozhranie

Do rozhrania potom vkladáme konštanty a hlavičky metód, napr.:

```
...  
public interface NazovRozhrania {  
    public int POCET = 20;           //private ani protected nie je dovolené  
    public double STAV = 4.77;  
    ...  
    public int vypocet(int cislo1,int cislo2); //bez tela  
    public void pocetZadani();  
    ...  
}
```

Rozhranie je potom pripravené na implementáciu ľubovoľnou triedou.

26.3 Implementácia rozhrania. Rezervované slovo implements.

Trieda môže implementovať jedno a viac rozhraní. Musí ale predefinovať všetky metódy rozhrania, inak kompilátor vyhlási chybu. Aby trieda mohla implementovať rozhranie musíme za jej názov uviesť rezervované slovo **implements** a za ním zoznam rozhraní, ktoré sa majú implementovať. V triede potom predefinujeme všetky metódy rozhrania:

```
public class nazovTriedy implements NazovRozhrania{  
    //v triede predefinujeme všetky metódy  
}
```

Trieda môže súčasne dediť od inej triedy a implementovať niekoľko rozhraní. Rozhrania majú pomerne široké využitie: rozhranie môžeme použiť ako typ referenčnej premennej, rozhrania možno dediť atď. My si ukážeme použitie rozhraní na príklade, v ktorom niekoľko tried bude implementovať po dve rozhrania.

26.4 Aplikácia PlosneObjekty.

Vytvoríme aplikáciu s názvom **PlosneObjekty**. Aplikácia bude obsahovať dve rozhrania: jedno, v ktorom budú metódy na popis plošného objektu a druhé, v ktorom budú metódy na výpočet obvodu a obsahu plošného objektu. Ďalej vytvoríme triedy Trojuholnik, Stvoruholnik a Kruh. Všetky triedy budú implementovať obe rozhrania a predefinujú metódy rozhraní podľa svojich potrieb. V triedach vytvoríme konštruktory a prístupové metódy. Na niekoľkých inštanciách potom vyskúšame funkčnosť metód.

26.5 Operátor instanceof.

Pri využívaní dedičnosti a rozhraní máme možnosť vybudovať si určitú hierarchiu tried. Táto hierarchia môže byť pomerne zložitá a preto môže byť niekedy problém určiť koho inštanciou je daný objekt. V tejto situácii môžeme využiť operátor **instanceof**.

26 Rozhranie

Pomocou operátora *instanceof* vieme zistiť, či je objekt inštanciou danej triedy:

```
nazovObjektu instanceof NazovTriedy
```

Tento výraz vráti *true*, ak je objekt inštanciou triedy. Ak nie je inštanciou triedy, tak vráti *false*.

26.6 Cvičenie

1. Vytvorte aplikáciu s názvom **PracaSCislami**. Vytvorte rozhranie *Vypocty* s metódami *sucet()*, *rozdiel()*, *sucin()*, *podiel()*. Deklarujte triedu *Cisla* s dvomi reálnymi premennými. Implementujte rozhranie. V triede predefinujte metódy, tak aby vykonávali príslušné operácie s premennými, ďalej vytvorte konštruktory a prístupové metódy.. Vytvorte inštanciu triedy a vyskúšajte funkčnosť metód.
2. Vytvorte ľubovoľnú aplikáciu, v ktorej bude trieda dediť od inej triedy a zároveň implementovať rozhranie s aspoň dvomi metódami.

26.7 Otázky

1. Aký je rozdiel medzi rozhraním a triedou?
2. Kedy je výhodné použiť rozhranie?
3. Čo je potrebné urobiť v triede, ktorá implementuje rozhrania?
4. Na čo slúži operátor *instanceof*?