

14. Dědičnost

Dědičnost v souvislosti se vztahem mezi objekty Předka a Potomka

- Vztah mezi objekty Předka a Potomka je založen na tom, že Potomek zdědí vlastnosti a metody z Předka a může také definovat své vlastní vlastnosti a metody
- Potomek může také přepsat (override) metody zděděné od Předka a poskytnout tak vlastní implementaci

Omezení dědičnosti (počet předků, souvislost s rozhraním)

- Omezení dědičnosti jsou spojena s počtem předků a rozhraním
- V Javě je možné, aby třída měla pouze jednoho předka (single inheritance). To znamená, že každá třída může být odvozena pouze od jedné jiné třídy. Toto omezení zajišťuje, že hierarchie tříd není příliš složitá a snadno se spravuje

Jazykové konstrukce (zápis) definice třídy Potomek odvozené od třídy Předek – extends, super

- Klíčové slovo **"extends"** se používá k definici třídy Potomek a určuje třídu Předek, ze které se třída Potomek odvozuje

```
public class Potomek extends PredkovaTrida {  
    // tělo třídy Potomek  
}
```

- Klíčové slovo **"super"** se používá uvnitř třídy Potomek a odkazuje na Předka. Může být použito k volání konstruktoru Předka nebo ke získání hodnoty zděděného atributu

```
public class Potomek extends PredkovaTrida {  
    public Potomek() {  
        super(); // volání konstrukturu Předka  
    }  
  
    public void necoDelam() {  
        int hodnota = super.getHodnota(); // získání hodnoty zděděného atributu  
    }  
}
```

Přístupová práva dědičnosti

- Public
 - Umožňuje přístup ke všem vlastnostem a metodám objektu, ať už jsou definovány kdekoli
- Private
 - Umožňuje přístup pouze z rámci třídy, ve které byly definovány
- Protected
 - Umožňuje přístup pouze v rámci třídy a tříd odvozených

- Default
 - pokud není specifikováno žádné přístupové právo, pak se jedná o default přístupové právo. Vlastnosti a metody s tímto přístupovým právem jsou viditelné pouze v rámci package, ve kterém byly definovány
-

Prakticky:

- Kód třídy, doplnit potomka, dědičnost v Javě u významných třídy (Výjimky, GUI)