

# Téma 10: Principy OOP (dědičnost, rozhraní, abstraktní třída)

---

V jazyce Java není možné dědit od více tříd najednou. Náhrada za toto jsou ale **rozhraní**. Interface nebo také rozhraní je speciální druh třídy. Deklarace rozhraní je podobná deklaraci třídy, akorát se použije klíčové slovo `interface` místo `class`.

```
public interface DataType {  
    public String getValue();  
    public String getDataType();  
    public void nullValue();  
    public void reset();  
}
```

Rozhraní definuje soubor metod, které ale v něm nejsou implementovány (v deklaraci rozhraní je pouze hlavička metody). Třída, která rozhraní implementuje, musí překrýt všechny jeho metody. Rozhraní nemůže deklarovat žádné proměnné, ale jen konstanty. Třída může implementovat více než jedno rozhraní. Při použití více než jednoho rozhraní, v hlavičce třídy deklarujeme rozhraní, přičemž nezáleží na pořadí. Rozhraní oddělujeme čárkami.

```
public class Trida implements DataType {  
    public String getValue() {  
        source code;  
    }  
    public String getDataType() {  
        source code;  
    }  
    public void nullValue() {  
        source code;  
    }  
    public void reset() {  
        source code;  
    }  
}
```

V programu nelze vytvářet instance rozhraní, ale mohu do proměnné typu vložit objekt typu, jenž implementuje dané rozhraní.

```
Interface_name variable_name = new Class_name(params);
```

Třída je **abstraktní třídou** právě tehdy, když obsahuje alespoň jednu abstraktní metodu, jež neobsahuje žádnou implementaci podobně jako u rozhraní. Hlavní rozdíl mezi abstraktní třídou a rozhraním je ten že v abstraktní třídě se můžou deklarovat proměnné a některé metody mohou obsahovat implementaci.

```
public abstract class DataType {  
    private int i;  
    public abstract String getValue();  
    public abstract String getDataType();  
    public abstract void nullValue();  
    public void reset() {  
        source code;  
    }  
}
```

Jak je vidět abstraktní metody musí po modifikátoru přístupu obsahovat také klíčové slovo `abstract`. Pokud metoda toto slovo neobsahuje, musí mít nějakou implementaci.

Pokud chci podědit abstraktní třídu, provádím to úplně stejně, jako když dědím obyčejnou třídu s tím rozdílem, že musím implementovat všechny abstraktní metody. Třída může dědit pouze jednu abstraktní třídu.

```
public class Trida extends DataType {  
    public String getValue() {  
        source code;  
    }  
    public String getDataType() {  
        source code;  
    }  
    public void nullValue() {  
        source code;  
    }  
}
```

V kódu není potřeba doprogramovat public void reset(), protože tato metoda je již naprogramována v abstraktní třídě.

Instanci abstraktní třídy opět nelze vytvořit, lze ovšem do proměnné abstraktní třídy vložit instanci třídy, která dědí od dané abstraktní třídy.