

Abstraktní třídy

Motivace

- Java disponuje rozhraními.
- Pak máme třídu(y) implementující určité rozhraní.
- Někdy je vhodné určité rozhraní implementovat pouze *částečně*:

Rozhraní

Specifikace

Abstraktní třída

Částečná implementace rozhraní (stačí mít hotové některé metody) a současně předek konkrétních tříd, tedy plných implementací

Neabstraktní třída

Úplná implementace rozhraní (musí mít hotové všechny metody)

Zápis abstraktní třídy

- Abstraktní třída je označena klíčovým slovem `abstract` v hlavičce, např.:

```
public abstract class AbstractSearcher
```

- Název začínající na `Abstract` není povinný ani nutný.
- Někdy se místo `Abstract` používá `Base`: `SearcherBase`, aby se zdůraznilo, že abstraktní třída je základem odvozených tříd konkrétních.
- Abstraktní třída má obvykle alespoň jednu *abstraktní metodu*, deklarovanou např.:

```
public abstract int indexOf(double d);
```

- Od abstraktní třídy *nelze vytvořit instanci*, (chybí implementace některých metod) nelze napsat např.:

```
Searcher ch = new AbstractSearcher(...);
```

Příklad: rozhraní → abstraktní třída → neabstraktní třída

Searcher

rozhraní — specifikuje, co má prohledávač umět

AbstractSearcher

abstraktní třída — předek konkrétních plných implementací prohledávače

LinearSearcher

konkrétní třída — plná implementace prohledávače

Searcher

`Searcher` je rozhraní = specifikuje, co má prohledávač umět

```
public interface Searcher {
    // Set the array for later searching
    void setData(double[] a);
    // Check whether array contains d element
    boolean contains(double d);
    // Return the position of d in the array (or -1 if not found)
    int indexOf(double d);
}
```

AbstractSearcher

`AbstractSearcher` je abstraktní třída = předek konkrétních plných implementací prohledávače

```
// this class implements Searcher only partially
public abstract class AbstractSearcher implements Searcher {
    // array, its getters and setters are implemented
    private double[] array;
    public void setData(double[] a) { array = a; }
    public double[] getData() { return array; }
    // we can call indexOf now though it will be implemented later
    public boolean contains(double d) {
        return indexOf(d) >= 0;
    }
    // finding the position of d is NOT implemented yet!
    public abstract int indexOf(double d);
}
```

LinearSearcher

`LinearSearcher` je konkrétní třída = plná implementace prohledávače, pomocí lineárního prohledání

```
public class LinearSearcher extends AbstractSearcher {
    // class has to implement all abstract methods
    public int indexOf(double d) {
        double[] data = getData();
        for(int i = 0; i < data.length; i++) {
            if(data[i] == d) {
                return i;
            }
        }
        return -1;
    }
}
```

Šablonové metody

- Všimněte si, že ve třídě `AbstractSearcher` volá metoda `contains` abstraktní metodu `indexOf`, která na této úrovni ještě neexistuje.
- Je to v pořádku, protože u kompletní třídy (viz `LinearSearcher` dále) již požadovaný kód musí být.
- Jedná se o návrhový vzor *Template Method (šablonová metoda)*, kdy kód třídy spoléhá na to, že kód šablonové metody dodají až podtřídy.

Repl.it demo k abstraktním třídám

- <https://repl.it/@tpitner/PB162-Java-Lecture-06-abstract-classes>