

13. Konstanty, Statické metody a vlastnosti, Anonymní metody, Ukazatel na metodu

Konstanty

Konstanty jsou neměnné hodnoty, které jsou známy v době kompilace a nemění se po dobu života programu.

Využití z konstantních hodnot je když víme, že se hodnota nebude měnit během života programu a umožní kompilátoru programu optimalizovat kód a zrychlit tím běh (o trochu, kompilátor vám uvěří, že se hodnota nebude měnit)

Klíčové slovo "const" se hlavně využívá, třeba u práce více lidí na jednom projektu, tak to slovo řekne ostatním, že se ta hodnota nikdy nemění

Musí mít vždy hodnotu, nemůže nastat situace, kdy bude null

Jsou tedy readonly

Konstanty jsou deklarovány s modifikátorem – final

Uživatelsky definované typy, včetně tříd, struktur a polí, nemůžou být const.

Statika

Nemusí být existovat instance nějaké třídy, aby se statický prvek dokázal zavolat.

Volá se na třídě.

Třeba Třída Console má statické metody WriteLine a Write atd...

OOP umožňuje definovat atributy a metody na samotné třídě. Tyto prvky jsou statické a jsou nezávislé na instanci.

Jde dát skoro před všechno: pole, metody, vlastnosti, operátory, eventy a konstruktory

V podstatě narušují objektový model. OOP je obsahuje jen pro speciální případy a obecně platí, že vše jde napsat bez statiky. Vždy musíme pečlivě zvážit, zda statiku opravdu nutně potřebujeme.

Statické vlastnosti

Statické prvky patří třídě

Data v nich uložená tedy můžeme číst bez ohledu na to, zda nějaká instance existuje.

Jsou společné pro všechny instance (i když s nima vůbec nesouvisí, ale nejde, aby instance měly rozdílné hodnoty u statických prvků)

```
class User {  
    public static int id = 0;
```

```

    private String name;
    public User(String name){
        this.name=name;
        id++;
    }
}

public static void main(String[] args){
    User u = new User("First");
    System.out.println(User.id); // 1
    User u = new User("Second");
    System.out.println(User.id); // 2
}

```

Statické metody

Statické metody se volají na třídě.

Jedná se zejména o pomocné metody, které potřebujeme často používat a nevyplatí se nám tvořit instanci.

Metody na Console třeba, nejde vytvořit její instance, protože konzole je jen jedna a bylo by zbytečné tvořit si z ní instanci

Další třeba metoda Round() ve třídě Math

Příklad

Pomocné statické metody na zkontrolování délky jména, aby mohla být instance v pořádku vytvořena

Delegát

Je to ukazatel na funkci (metodu) = uložení metody do proměnné

Při jeho vytváření je nutné určit návratový typ a vstupní parametry metody. obsahuje referenci metody

objektově orientované a typově bezpečné ukazatele na funkce. Je možné deklarovat funkci, která bude mít jako parametr delegáta.

Existuje i multi-část delegate, když se mu přidávají více metod +=

Event

Podobný delegátu, taky slouží pro datovou komunikaci k jeho vytvoření je zapotřebí delegáta

Anonymní metody

vytvoření dočasné proměnné

je "vložený" příkaz nebo výraz, který lze použít všude, kde je očekáván typ delegáta

```
String[] arr = { "xxx", "cd", "ab", "z" };
Arrays.sort(arr, new Comparator<String>() {
    @Override public int compare(String s1, String s2) {
        return s1.length() - s2.length();
    }
});
System.out.println(Arrays.toString(arr));
// prints "[z, cd, ab, xxx]"
```