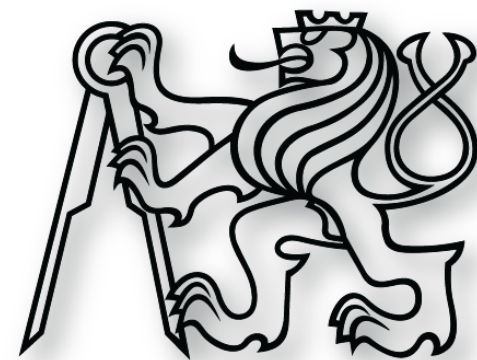


# Objektově orientované programování

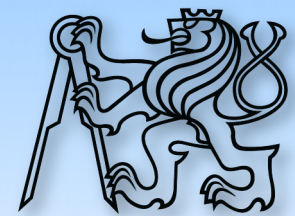
cvičení 3. března 2014



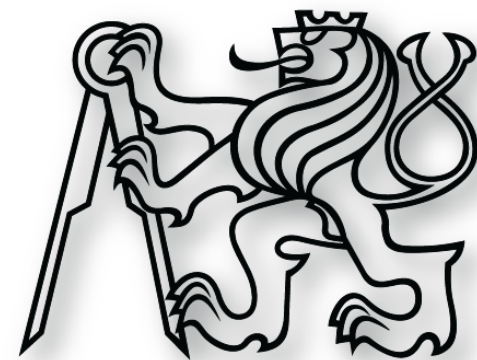


# Opakování teorie

# Mluvili jsme o ...

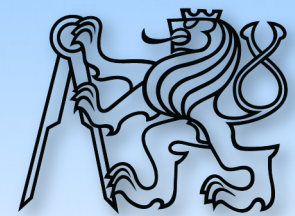


- Proměnná
- Pole
- Datová primitiva
- Deklarace
- Inicializace
- Viditelnost (scope) proměnné
- Rekurze
- Signatura metody



# Opakování praxe

body za aktivitu



## Určete výsledek programu

**A**

```
class Main {
    static int cislo = 1;

    public static void main() {

        if (cislo > 1) {
            int cislo = 2;
            cislo *= 3;
        } else {
            cislo *= 5;
            int cislo = cislo;
        }
        System.out.println(cislo);
    }
}
```

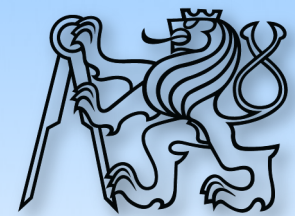
**B**

```
public static void main() {

    boolean isABC = true;

    if (isABC) {
        String text = "ABC";
    } else {
        String text = "DEF";
    }

    System.out.println(text);
}
```



## Určete výsledek programu

```
public static void main() {  
    String text = "ABC";  
  
    if (text == "ABC") {  
        boolean isABC = true;  
    } else {  
        boolean isABC = false;  
    }  
  
    System.out.println(isABC);  
}
```

A

```
class Main {  
    static int cislo = 3;  
  
    public static void main() {  
  
        if (cislo <= 3) {  
            int cislo = cislo;  
            cislo *= 7;  
        } else {  
            cislo *= 2;  
            int cislo = 1;  
        }  
        System.out.println(cislo);  
    }  
}
```

B

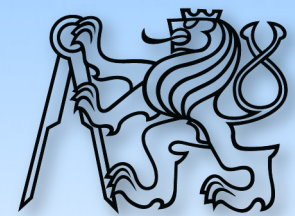


**Rekurze: Implementujte funkci pro výpočet n-tého členu Fibonacciho posloupnosti**

1, 1, 2, 3, 5, 8, 13, 21, ...

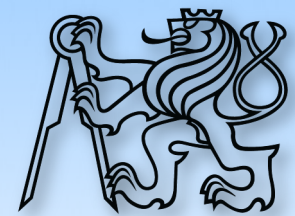
```
static int fib(int poradi) {  
  
}
```

# Odevzdání



- Emailem na [cemuskar@fel.cvut.cz](mailto:cemuskar@fel.cvut.cz)
- Za 2 ze 3 jeden bod za aktivitu





## Určete výsledek programu

**A**

```
class Main {
    static int cislo = 1;

    public static void main() {

        if (cislo > 1) {
            int cislo = 2;
            cislo *= 3;
        } else {
            cislo *= 5;
            int cislo = cislo;
        }
        System.out.println(cislo);
    }
}
```

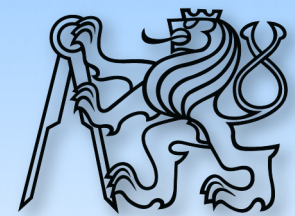
**B**

```
public static void main() {

    boolean isABC = true;

    if (isABC) {
        String text = "ABC";
    } else {
        String text = "DEF";
    }

    System.out.println(text);
}
```



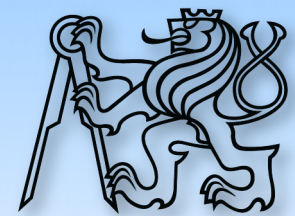
## Určete výsledek programu

```
public static void main() {  
    String text = "ABC";  
  
    if (text == "ABC") {  
        boolean isABC = true;  
    } else {  
        boolean isABC = false;  
    }  
  
    System.out.println(isABC);  
}
```

A

```
class Main {  
    static int cislo = 3;  
  
    public static void main() {  
  
        if (cislo <= 3) {  
            int cislo = cislo;  
            cislo *= 7;  
        } else {  
            cislo *= 2;  
            int cislo = 1;  
        }  
        System.out.println(cislo);  
    }  
}
```

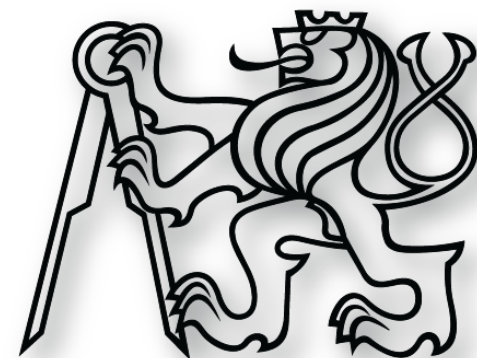
B



## Rekurze: Implementujte funkci pro výpočet n-tého členu Fibonacciho posloupnosti

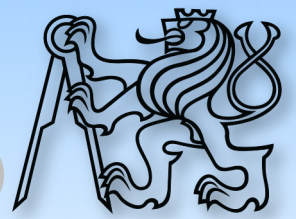
1, 1, 2, 3, 5, 8, 13, 21, ...

```
static int fib(int poradi) {  
    if (poradi == 1 || poradi == 2) return 1;  
    else return fib(poradi-1) + fib(poradi-2);  
}
```



# Objektově orientovaný přístup

vysvětlení + příklady



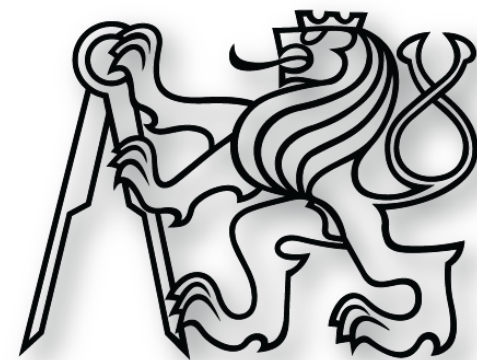
- Objektově orientovaný
  - přístup
  - programování
  - paradigma
- Svět (vše) se skládá z objektů
  - → snadné pro představivost
- Hmotné i nehmotné
- Objekty jsou instancemi šablon
- Každý objekt má
  - vlastnosti (atributy)
  - operace (metody)
  - šablonu (nebo více)
- Hierarchie šablon

# Třída vs. Objekt



- Třída = vzor, šablona, předpis
- Datový typ
- Definiuje
  - Účel a charakter
  - Odpovědnost
  - Vlastnosti
  - Akce
- Tvoří hierarchii
  - Postupné zpřesňování
- Z 1 šablony neomezeně instancí
- Instance právě 1 třídy
- Reprezentuje právě 1 objekt
- Vlastní místo v paměti
  - Na haldě

PJW



**S chutí do praxe**