

Co je to programování?

PLIN048 – Základy programování pro humanitní obory

Richard Holaj
FF MU

28. února 2017

Co nás dnes čeká?

Co je uloženo v paměti?

Princip datových typů

Jak určit datový typ?

Přetypování a typová omezení

Složitější strukturování paměti

Možnosti strukturování

Uspořádané a neuspořádané kolekce

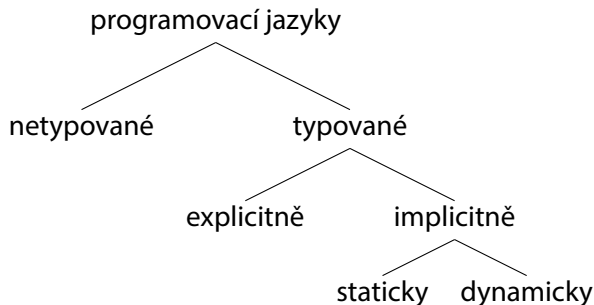
Reprezentace složených proměnných v paměti

Cvičení

Princip datových typů

- ▶ různé typy hodnot – čísla celá a reálná, text, ...
- ▶ dáno programovacím jazykem
- ▶ rozsah hodnot (velikost)

Typologie jazyků podle způsobu typování



Ukázky v různých jazycích

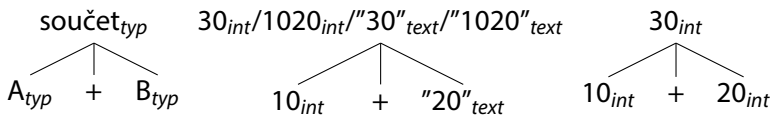
Java (explicitní statické)

```
1 | int a = 42;  
2 | String b = "Hello universe!";  
3 | boolean c = true;  
4 | float d = 10.0;
```

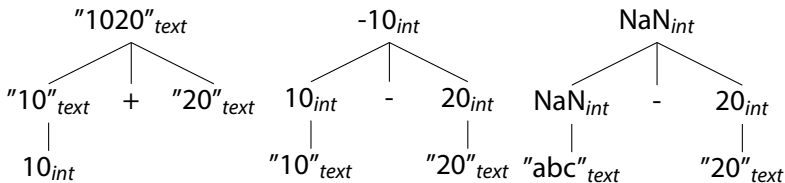
Python (implicitní dynamické)

```
1 | a = 42  
2 | b = "Hello universe!"  
3 | c = True  
4 | d = 10.0
```

Typová omezení operací



Obrázek: Omezení pro operaci součet, kdy $\text{typ} \in \{\text{int}, \text{text}, \dots\}$



Obrázek: Operace součet a rozdíl v JavaScriptu (automatické přetypování)

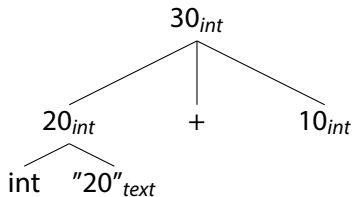
Explicitní přetypování

Zápis v Pythonu

```
1 | int("20") + 10
```

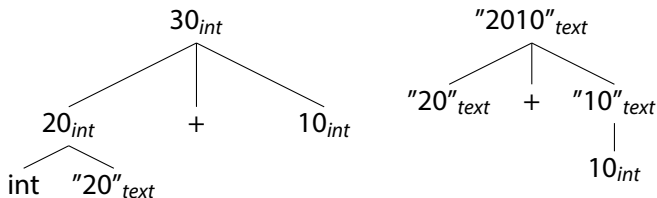
Zápis v Javě

```
1 | (int) "20" + 10
```



Obrázek: Vyhodnocení v Pythonu

Explicitní vs. implicitní přetypování



Obrázek: Explicitní přetypování vs. implicitní přetypování

Možnosti strukturování

- ▶ Kolekce
 - ▶ Uspořádané
 - ▶ Neuspořádané
- ▶ Objekty – později

Uspořádané a neuspořádané kolekce

- ▶ Společné vlastnosti
 - ▶ položky stejného vs. různého typu
 - ▶ zanořování
- ▶ Uspořádané
 - ▶ seznamy, pole
 - ▶ [], indexy
- ▶ Neuspořádané
 - ▶ slovníky, „objekty“, mapy
 - ▶ {}, klíče

Uspořádané a neuspořádané kolekce – ukázka

```
1 seznam = [10, 20, "AHOJ", ["A", 20], {"TEST": 42}]
2 print(seznam[2]) # "AHOJ"
3 seznam[2] = 751
4 print(seznam[2]) # 751
5 print(seznam[3][1]) # 20
6 seznam[3][1] = "B"
7 print(seznam[3][1]) # "B"
8 print(seznam[4]["TEST"]) # 42
9 seznam[4]["TEST"] = 42 * 2
10 print(seznam[4]["TEST"]) # 84
11 slovník = {"A": "a", "l": [1, 3], "hex": {"FF": 256}}
12 print(slovník["A"]) # "a"
13 slovník["A"] = 42
14 print(slovník["A"]) # 42
15 print(slovník["l"][0]) # 10
16 slovník["SEZNAM"][0] = [1, 2]
17 print(slovník["l"][0]) # [1, 2]
18 print(slovník["l"][0][0]) # 1
19 print(slovník["hex"]["FF"]) # 256
```

Reprezentace složených proměnných v paměti – reference vs. hodnota

Obrázek: Proměnná jako hodnota

$$a = [10, 20]$$



$$b = a$$

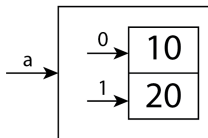


$$b = 7$$

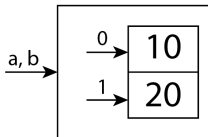


Obrázek: Proměnná jako reference

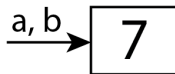
$$a = [10, 20]$$



$$b = a$$

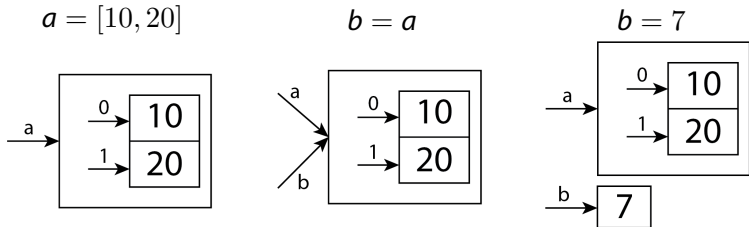


$$b = 7$$

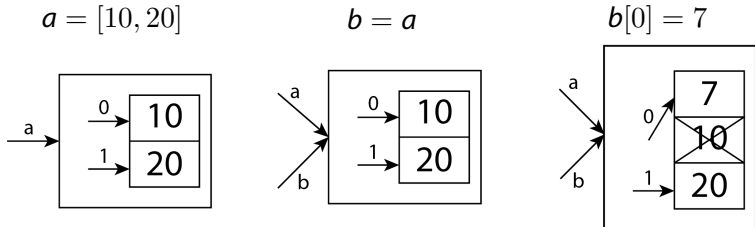


Hybridní reprezentace hodnota/reference

(a) Přiřazení hodnoty kolekci



(b) Přiřazení hodnoty položce



Cvičení

Jaký je výsledek tohoto kódu v jazyce Python? Nakreslete strom vyhodnocení výrazu.

```
1 | a = "10"  
2 | b = 20  
3 | c = a + b
```

Jaký je výsledek tohoto kódu v jazyce Python? Nakreslete strom vyhodnocení výrazu.

```
1 | a = "10"  
2 | b = 20  
3 | c = int(a) + b
```

Cvičení

Jaký je výsledek tohoto kódu v jazyce JavaScript? Nakreslete strom vyhodnocení výrazu.

```
1 |   var a = "10"  
2 |   var b = 20  
3 |   var c = a + b
```

Jaký je výsledek tohoto kódu v jazyce JavaScript? Nakreslete strom vyhodnocení výrazu.

```
1 |   var a = "10";  
2 |   var b = 20;  
3 |   var c = parseInt(a) + b;
```

Cvičení

Napište příkaz, který v jazyce Python vypíše hodnotu 42 nacházející se v kolekci (proměnné) *data*.

```
1 | data = {"A":72,B:[42, 10]}
```

Napište příkaz, který v jazyce Python vypíše hodnotu 42 nacházející se v kolekci (proměnné) *data*.

```
1 | data = [[[10, [[42]]]]]
```


Cvičení

Jaké jsou hodnoty proměnných a a b po provedení následujícího kódu, jedná-li se o proměnné typu hodnota?

```
1 | a = {"A": [10, 17], "B": 42}
2 | b = a
3 | b["A"][1] = 3
```

Jaké jsou hodnoty proměnných a a b po provedení následujícího kódu, jedná-li se o proměnné typu reference?

```
1 | a = {"A": [10, 17], "B": 42}
2 | b = a
3 | b["A"][1] = 3
```

Jaké jsou hodnoty proměnných a a b po provedení následujícího kódu, jedná-li se o proměnné hybridního typu hodnota/reference (např. v Pythonu nebo JavaScriptu)?

```
1 | a = {"A": [10, 17], "B": 42}
2 | b = a
3 | b["A"][1] = 3
```