



# Programování

Práce s pamětí



# Referenční a hodnotové datové typy

## ➤ **Hodnotové**

- Numerické (int, long, double, ...)
- Logické (bool)
- Textové (char, string)
- Výčtový typ (enum)
- Struktury

## ➤ **Referenční**

- Objekty



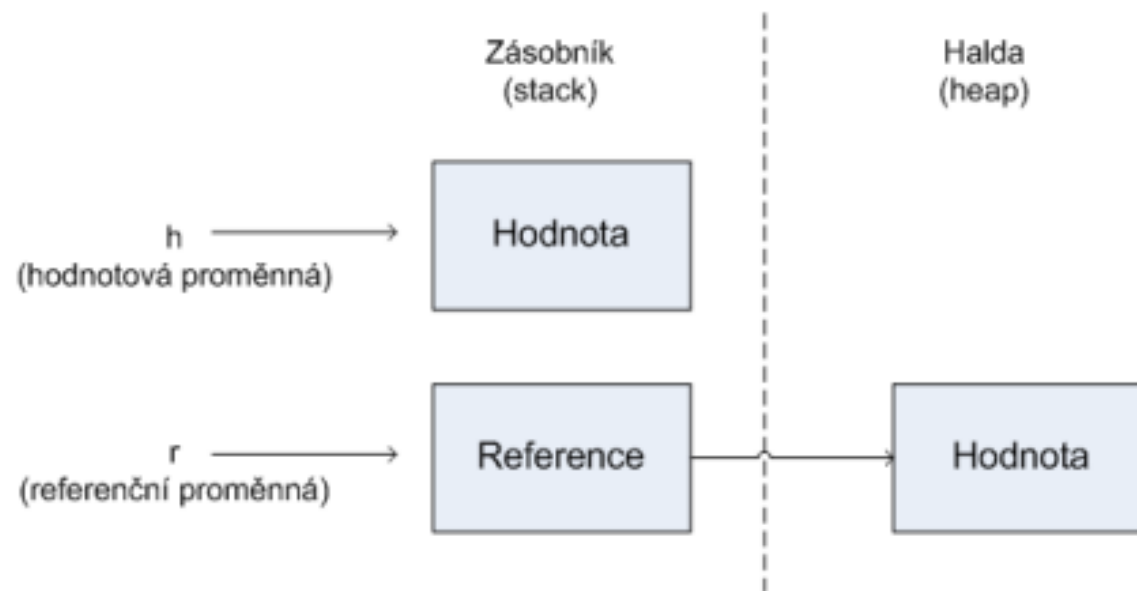


# Ukládání do paměti

- Hodnotové datové typy jsou ukládány přímo na zásobník (LIFO)
- Data jsou uložena přímo v paměti, na kterou odkazuje proměnná
- Referenční datové typy mají hodnotu uloženou na haldě (heap)
- V zásobníku je uložen pouze referenční odkaz na haldu
- Každá aplikace má vyhrazenou kapacitu zásobníku
- Pro praktickou činnost není samotné uložení proměnných zásadní
  - Je ovšem třeba mít tento fakt na paměti

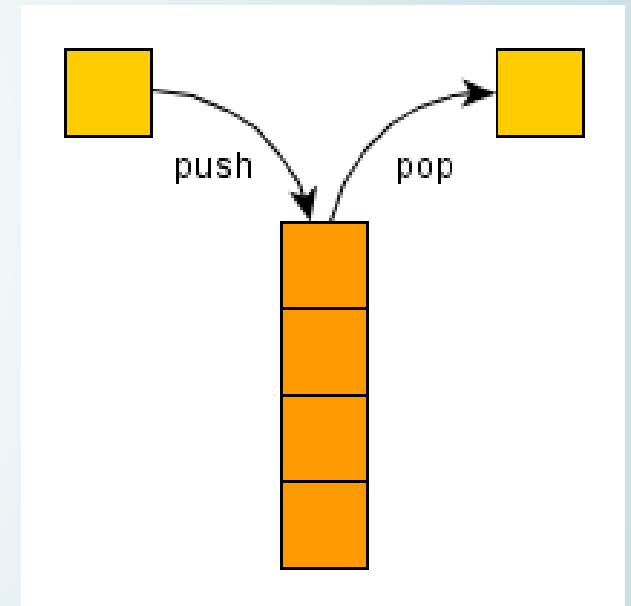
# Ukládání do paměti

- S využitím adresace (pointer) lze mít více referenčních proměnných ukazujících na stejnou hodnotu
- Reference tak zůstává odkazovat na stejnou část v paměti, ale hodnota se může měnit
- O správu paměti, zejména její uvolňování se stará **GARBAGE COLLECTOR**



# Zásobník (stack)

- Datová struktura pro dočasné ukládání paměti
- Každý námi vytvářený program má pevně stanovenou velikost zásobníku
- Poslední uložená položka do paměti je první vybrána (LIFO)
- Zásobník má definované dno (prázdný zásobník) a vrchol
- Pro práci se zásobníkem se využívají funkce **push** (vložení) a **pop** (vyjmutí)



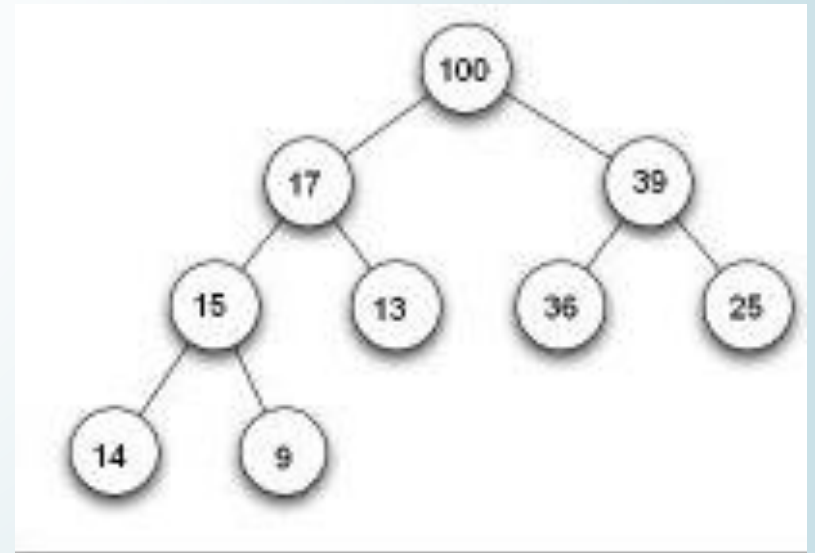
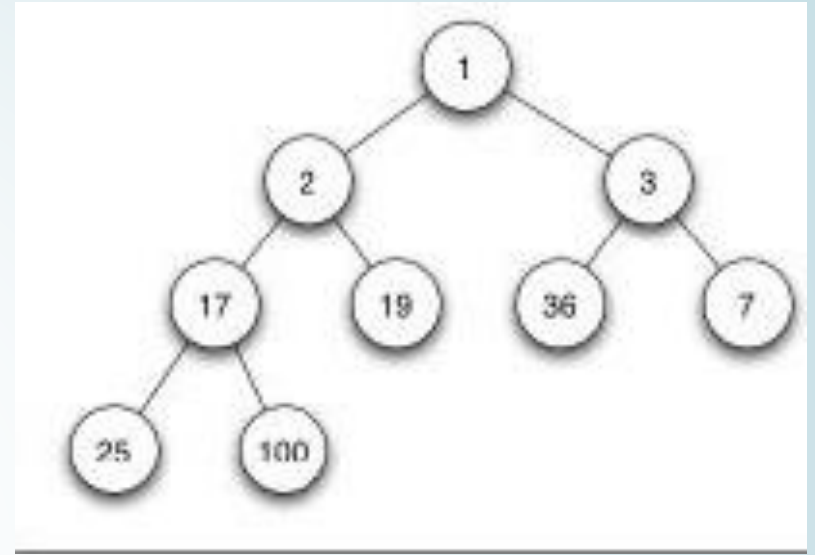


# Halda (heap)

- Vkládáme na první volnou pozici – následně dochází k úpravě haldy
- Jednotlivé vložené prvky se nazývají uzly
- Rozlišujeme maximální haldu a minimální haldu
- Rozdíl je zda je v kořenu nejnižší nebo nejvyšší hodnota

# Minimální a maximální halda

- Každý nově vkládaný prvek vložíme na první volné místo (postupujeme postupně do hloubky)
- Začínáme přiřazením pod kořen
- Samotné umístění prvku závisí o který typ haldy (MIN/MAX) se jedná
- Po vložení prvku je halda postupně upravována tak, aby odpovídala MIN/MAX haldě



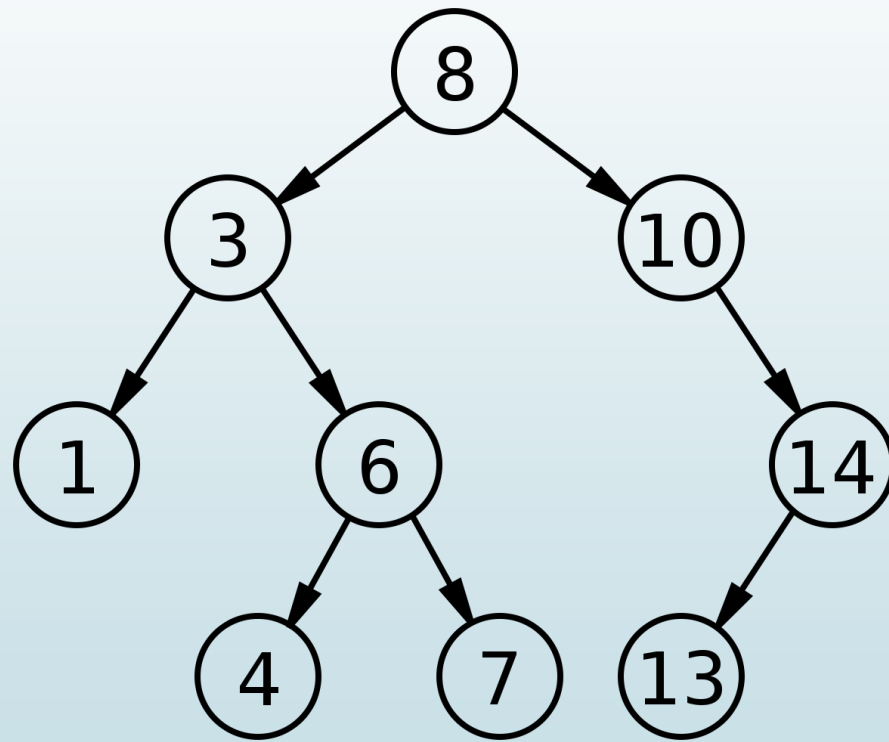


# Binární vyhledávací strom (BST)

- **Binární stromová** datová struktura
- Na vrcholu je tzv. **kořen stromu**
- Nové hodnoty se vkládají do tzv. **listů**
- Oproti haldě se nově vložené prvky umisťují na první volnou pozici dle pravidla vyhledávacího stromu
- Nově vkládanou hodnotu postupně porovnáváme s hodnotou v uzlu
  - Pokud je hodnota menší, pokračujeme na levý podstrom
  - Pokud je hodnota větší rovna, pokračujeme na pravý podstrom



# Binární vyhledávací strom (BST)



- V kořenu je první vložená hodnota
- Vyhledávací strom není vyvážený oproti haldě