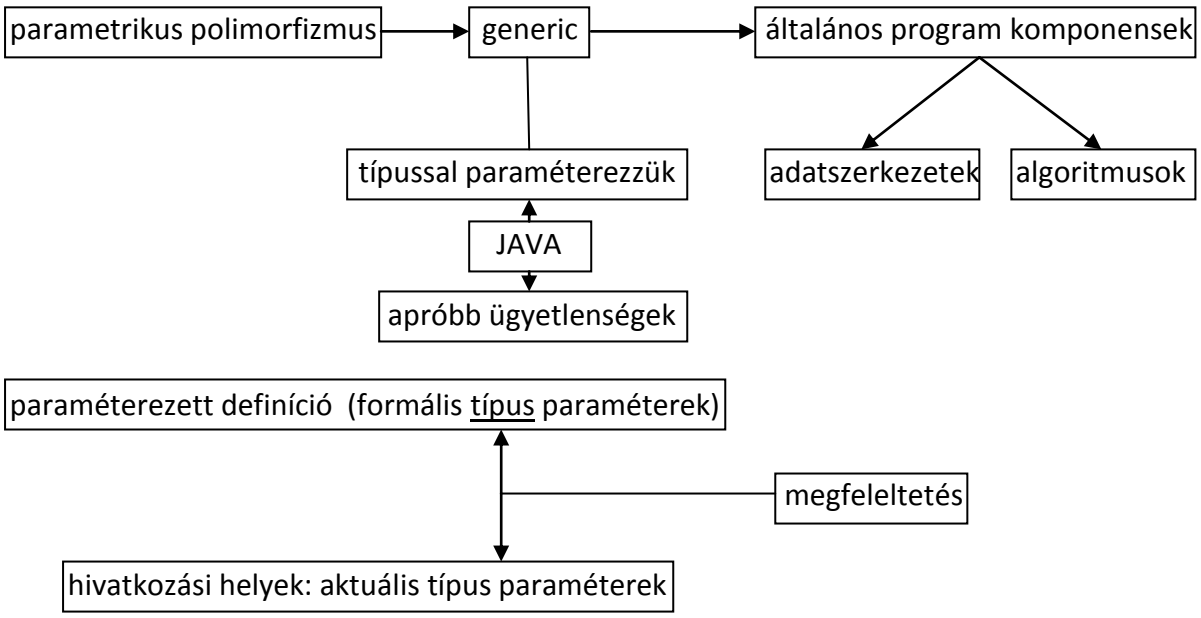
Programozási nyelvek Java

Kozsik Tamás előadása alapján
Készítette: Nagy Krisztián

10. előadás

Polimorfizmus

- **ad-hoc polimorfizmusok**
  - túlterhelés(például operátoroknál)
  - automatikus típuskonverziók
- **univerzális polimorfizmusok**
  - altípusos polimorfizmus
  - parametrikus polimorfizmus



Tömb: beépített adatszerkezet.
- problémája: fix méretű adatszerkezet. Amennyiben elfogy az előre megadott hely, úgy az új mérettel egy újat kell allokálni és átmásolni a kezdetleges tömb elemeit (+ overhead)

Fontos előnye: nem csak referencia típusokat lehet benne tárolni, hanem primitív típusokat is! Például: int

ArrayList: beépített generikus adatszerkezet.
- problémája csak referencia típus lehet aktuális típus paraméter!
- primitív kiváltható csomagoló osztályból (+ overhead)

Például: ArrayList<String> ArrayList<Integer> ~~ArrayList<int>~~

Példa ArrayList használatára:

```
ArrayList<String> a = new ArrayList<String>();
a.add(”Pityu”);
String str = a.getFirst();
```

Minta adatszerkezet: Verem

```
package datastruct;

public class Stack<T>{

     T[] data = new T[100];   // nem működik! (Ez is ügyetlenség)

      T value;
      Stack<T> next;
      …
}
```

ColoredPoint <: Point (class ColoredPoint extends Point…)

ArrayList<ColoredPoint> ~~<~~: ArrayList<Point>

ColoredPoint[]<: Point[]

Miért lenne baj ArrayList esetén a „kompatibilitással”?

```
ArrayList<ColoredPoint> cps = new ArrayList<ColoredPoint>();
cps.add(new ColoredPoint());
ColoredPoint x = cps.get(0);

ArrayList<Point> ps = cps;
ColoredPoint y = ps.get(0);   - Fordítási hiba!
ColoredPoint z = (ColoredPoint) ps.get(0);-  értelmes!
ps.add(new Point()); - lefordul, de butaság!
```

Nyílt nap miatt téma váltás!

Kifejezések kiértékelése

- Lexikális szabályok (Miből épül fel?)
- Szintaktikus szabályok (Hogyan épül fel)
- Szemantikus szabályok (Mi a jelentése? Hogyan értelmezzük?)

Lexika

- változó
- függvényhívás
- operátor
- zárójel
- vesszők
- literál (3.14, ”pi”, tömb literálok)

Szintaxis

- operátorok
  - aritás (hány argumentuma van)
    - 1 argumentum: ++
    - 2 argumentum: +
    - 3 argumentum: ? :
  - fixitás
    - prefix pl.: ++c
    - posztfix pl.: c++
    - infix pl.: a+b
    - mixfix pl a?b:c
  - függvények
    - vararg (változó hosszúságú paraméterlista)

Szemantika

- precedencia $(3 + 2 * 4 \approx 3 + (2 * 4))$
- asszociativitás $3 + 2 + 4 \rightarrow 3 + (2 + 4)\ vagy\ (3 + 2)$
  - bal asszociatív pl.: + $3 + a + b = (3 + a) + b$
  - jobb asszociatív pl.: = $x = y = 1 \rightarrow x = (y = 1)$
- mellékhatásosság (JAVA nem tiszta nyelv)
  - kifejezés
    érték (kiértékelés) mellékhatás (változók változnak)

    x++ ++x mellékhatása ugyan az. Érték szerint: $x + + \equiv x + + x \equiv x + 1$