Specifikacija, Implementacija & Abstrakcija

Specifikacija S $\qquad\qquad$ $\{P\}\ C\ \{Q\}$

zahteva (opis), kaj želimo

Implementacija I

izdelek $\qquad\qquad$ I **zadošča** S, če ustreza pogojem S

Algebra: vektorski prostor, grupa, monoid, kolobar } algebrske strukture

Definicija:

- **signatura**: množice, elemente, operacije
- **aksiomi**: logični zakoni, ki jim morajo elementi & operacije zadoščati

Vektorski prostor:

- **signatura**:

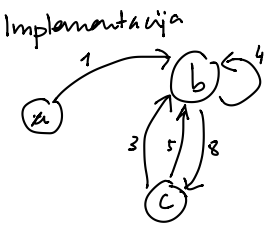
  V množica
  $\vec{0} \in V$ ničelni vektor
  $+ : V \times V \to V$
  $\cdot : \mathbb{R} \times V \to V$

  } sestavni kosi

- **aksiomi**: $\vec{0} + \vec{x} = \vec{x}$ $\qquad$ $1 \cdot \vec{x} = \vec{x}$
  $\vec{x} + \vec{y} = \vec{y} + \vec{x}$
  $\vdots$

  } funkcionalnost "kaj delajo kosi"

Usmerjen graf



$V = \{a, b, c\}$

$E = \{1, 3, 5, 8, 4\}$

	src	trg
1	a	b
3	c	b
5	c	b
8	b	c
4	b	b

signatura:

množica V (vozlišča)

množica E (povezave)

src : E → V

trg : E → V

aksiomi: ni aksiomov

Uporaba specifikacij:

1) zahteva ali opis, kaj naj koda počne (specifikacija)
2) protokol za uporabo kode (vmesnik / interface)

API application programming interface

Java

implementacija → class Foo { <implementacija> } ... datoteka Foo.java

vmesnik → interface

Ocaml

- implementacija → modul/struktura

1. module I =
   struct
   ⋮ (implementacija)
   end

2. V datoteko foo.ml damo implementacijo
   → modul Foo

- specifikacija → signatura

1. module type S =
   sig
   ⋮ (specifikacija)
   end

2. V datoteko foo.mli damo specifikacijo
   → specifikacija za foo.ml

Kako povemo, da I ustreza S?

Java:

```
public class C implements I,J,K {
    ⋮
}
```

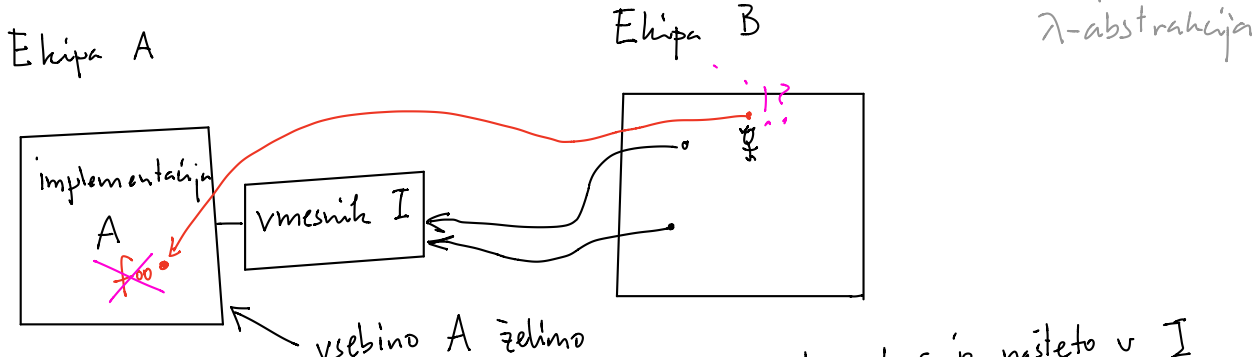
Ocaml:

1.
```
module C : I =
struct
  ⋮
end
```

Abstrakcija (Abstraction)

šef

To ni $\lambda x.x$ (↑ λ-abstrakcija)

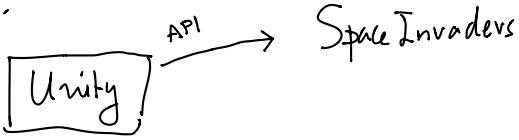


vsebino A želimo "skriti", da bo vidno samo to, kar je našteto v I

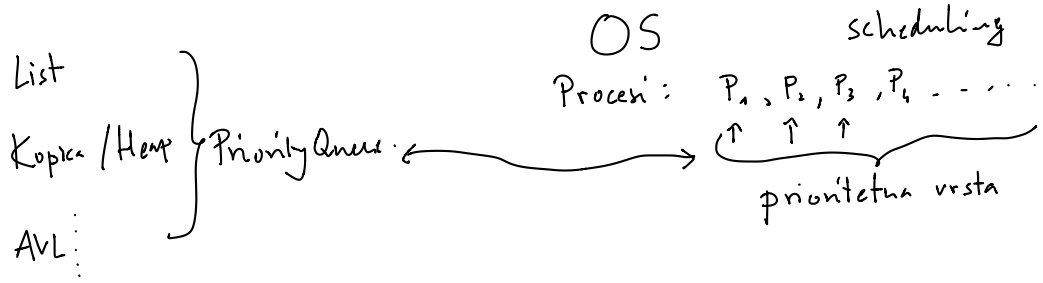
Abstrakcija

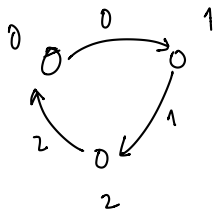
Generično programiranje

- Preproste specifikacija: Implementiraj Space Invaders za PS4.
- V resnici:



Primer:


List
Kopica / Heap
AVL
PriorityQueue
OS
scheduling
Procesi: $P_1, P_2, P_3, P_4 \ldots$
prioritetna vrsta


0
1
2
0
1
2
0
1
2