

Specifikacija & implementacija

Specifikacija : zahteva, želja, opis S

Implementacija : izdelek I

Ali I zadošča / ustreza S?

Signature & teorije v algebri

Grupa :

SIGNATURA (sestavni deli) {

- množica G
- element $e \in G$ ("enota")
- operacija $m: G \times G \rightarrow G$ ("množenje")
- operacija $i: G \rightarrow G$ ("inverz")

-
-1

AKSIOMI (lastnosti sestavnih delov) {

Aksiomii:

1. $m(x, m(y, z)) = m(m(x, y), z)$ $x \cdot (y \cdot z) = (x \cdot y) \cdot z$
2. $m(x, e) = x = m(e, x)$
3. $m(x, i(x)) = e = m(i(x), x)$

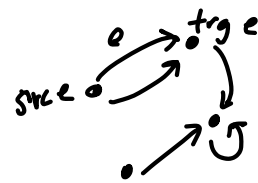
TEORIJA = SIGNATURA + AKSIOMI

Primer: geometrija

signatura : točke, premice,
premica (T_1, T_2)
sečišče (p_1, p_2)
⋮

aksiomi : aksiomi ravninske geometrije

Primer: usmerjeni graf



Signatura:

- V - množica "vozlišč"
- E - množica "povezav"

$$\text{src} : E \rightarrow V$$

$$\text{trg} : E \rightarrow V$$

aksiomi: ni aksiomov

Specifikacije v prog. jeziku

vmesnik (interface)

API (application programming interface)



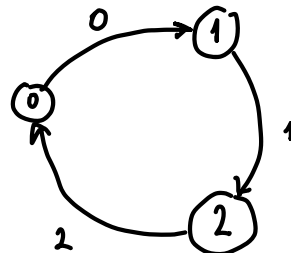
Specifikacija: opis zahtev

Opis uporabe (kako uporabimo knjižnico)

```
module Cycle3 : DIRECTED_GRAPH =  
struct  
  type v = int (* uporabimo 0, 1, 2 *)  
  type e = int (* uporabimo 0, 1, 2 *)  
  let src e = e  
  let trg e = (e + 1) mod 3
```

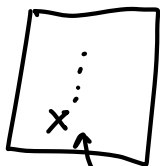
$$\text{src } 0 = 0$$
$$\text{trg } 0 = 1$$

$$\text{src } 2 = 2$$
$$\text{trg } 2 = (2 + 1) \bmod 3 = 0$$



Cycle(n)

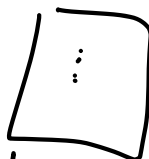
foo.ml



bar.ml
Foo.x

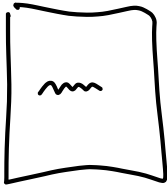


module Foo =
struct

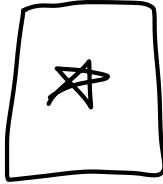


end

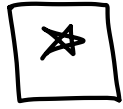
bat.ml



bat.mli



module Bat : sig



end

= struct



end

module F(S : I) =

⋮

class F <S extends I> {

⋮

}