Uso di un **contratto** per la

Correttezza del SW

Introduzione

H1

Abbiamo visto che il Design by Contract ci porta a progettare il codice seguendo un **contratto**, che coincide con la **specifica**, che può essere scritta in diversi linguaggi.

H2

Il contratto può essere usato come:

- **Documentazione**
- **Controllo runtime**
  - Le condizioni possono essere attivate o disattivate in runtime
    - attivare durante il testing
    - disattivate nel codice sul mercato (rallentano l'esecuzione)
- **Prova di *correttezza* del SW**
  - dimostrare che il contratto è rispettato dall'implementazione

Vediamo quindi questo terzo punto, assumendo per ***correttezza*** la definizione data qui .

Correttezza di un metodo

H2

- A : istruzioni o metodo
- P : precondizioni di A
- Q : postcondizioni di A
- **correttezza**: $\{P\}A\{Q\}$
  - ogni esecuzione di A che inizi in uno stato che soddisfa P termina in uno stato che soddisfi Q

Correttezza di una classe

Nel DbC una classe è corretta se sono corretti:

H2

- **operazione di creazione**
  - $\{PRE_{\text{constr}}\}\text{constructor}\{INV \land POST_{\text{constr}}\}$
- **ogni altro metodo**
  - se chiamo un metodo `operation` con PRE vero e INV vero allora dopo l'esecuzione del metodo vale ancora INV e vale $POST$:

    $\{PRE_{\text{op}}\}\text{operation}\{POST_{\text{op}} \land INV\}$