9.3 – Template e header

Libro di testo:

- Capitoli 19.3, 19.3.1, 19.3.2, 19.3.3*, 19.3.5 19.3.6

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

MEMENTO

- Si includono solo i file header (`.h`), non i file sorgente (`.cpp`)

Rational.h

```
class Rational {
  public:
      Rational(int num, int den);
      int numerator(void) const;
      int denominator(void) const;
      //...

  private:
      int numerator_;
      int denominator_;
};

bool operator==(Rational a, Rational b);
```

Rational.cpp

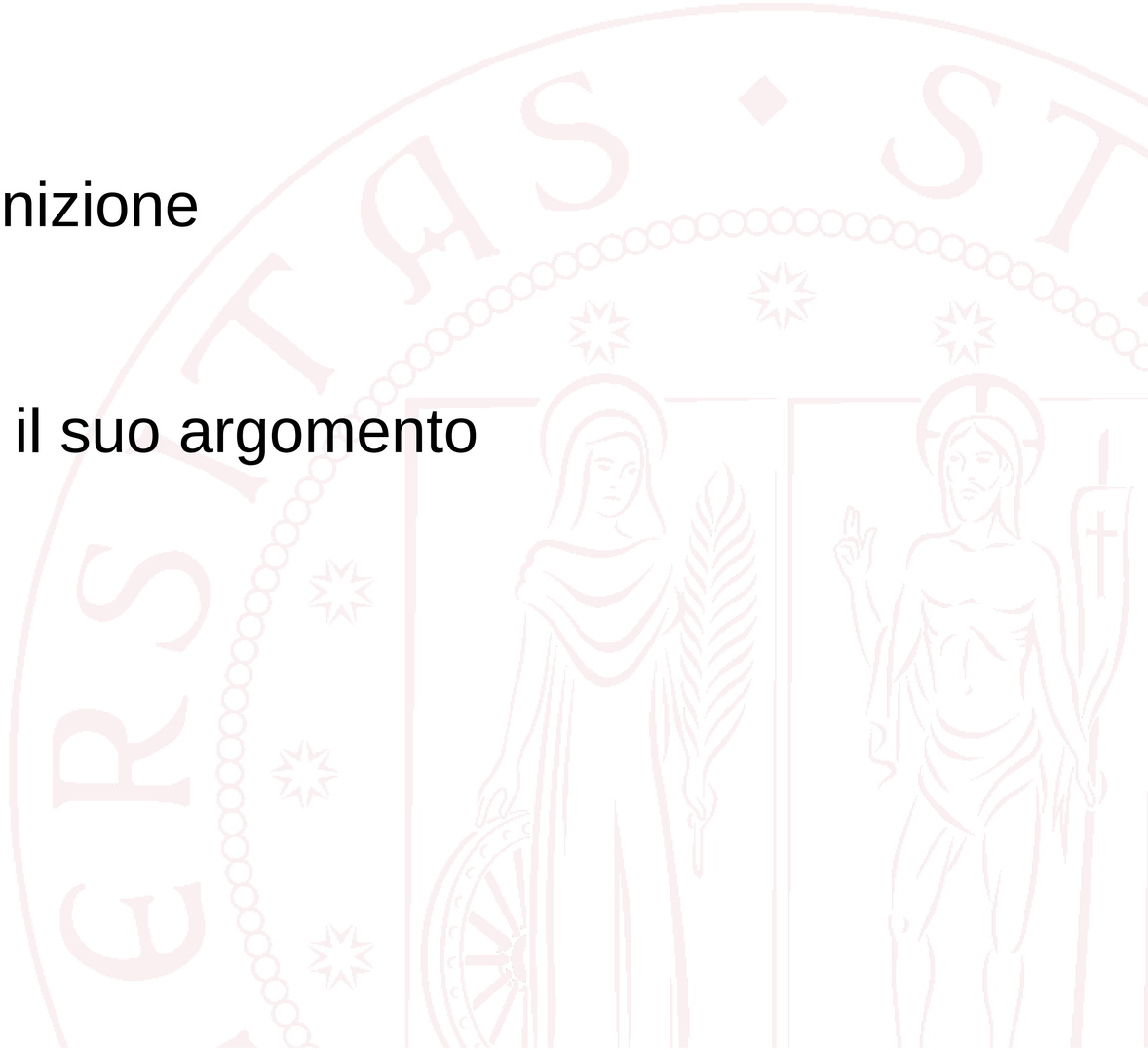
```
#include "Rational.h"
#include "Rational.cpp"

Rational::Rational(int num, int den)
    : numerator_{num}, denominator_{den} {
    //...
}

//...

bool operator==(Rational a, Rational b) {
    //...
}
```

Agenda

- Uso dei template nei file header
- Gestione
- Ripercussioni

Effetti collaterali

- I template combinano flessibilità e efficienza
- Effetti collaterali
  - Poca separazione tra dichiarazione e definizione
  - Errori di compilazione poco comprensibili
  - Il compilatore controlla sia il template che il suo argomento



Effetti collaterali

- I compilatori spesso richiedono che i template siano completamente definiti prima di essere usati
  - Le definizioni sono spostate negli header!
- Non è imposto dallo standard, ma dallo specifico compilatore

Header con i template

- Come mantenere la separazione tra interfaccia e implementazione con i template?

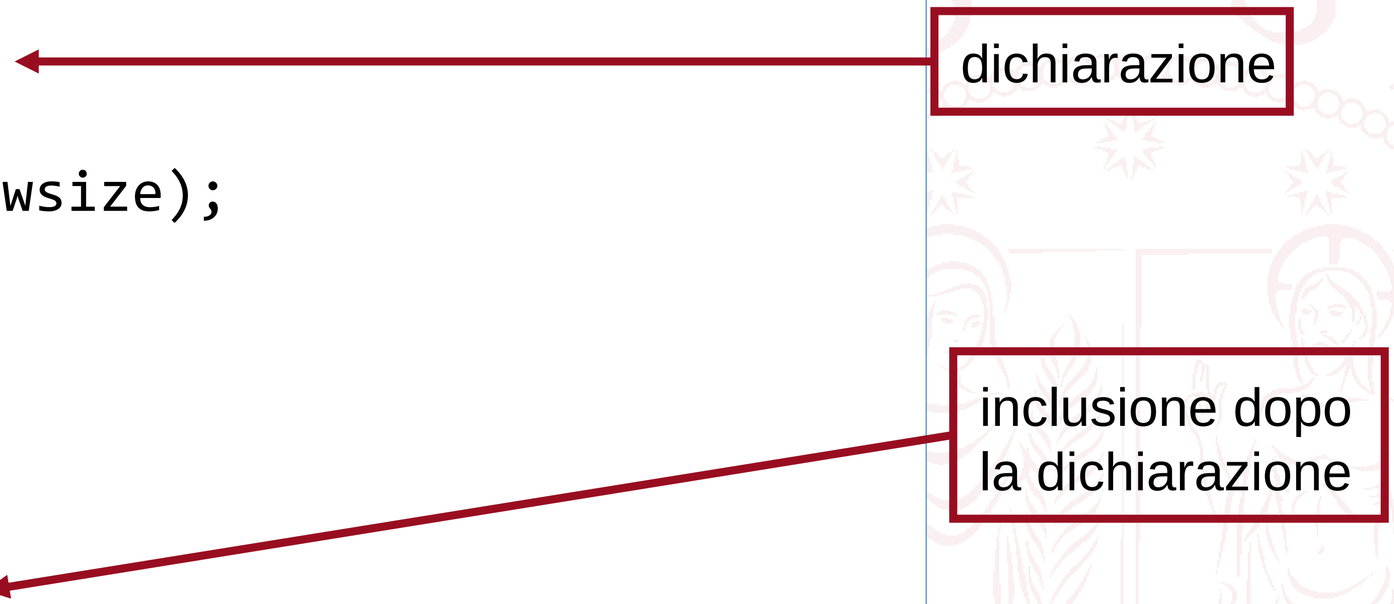
Header con i template

- Come mantenere la separazione tra interfaccia e implementazione con i template?
- L'header viene suddiviso in due blocchi:
  - **File `.h`** – tipico contenuto da header:
    - Definizioni di classi
    - Dichiarazioni di funzioni
  - **File `.hpp`** – tipico contenuto da `.cpp`
    - Definizioni di funzioni
- I blocchi sono combinati tramite un `#include` atipico

Header con i template

vector.h

```
#ifndef vector_h
#define vector_h

template<typename T>
class vector{
// ...
    void resize(int newsize);
// ...
};

// ...

#include "vector.hpp"

#endif // vector_h
```



Header con i template

vector.hpp

```
#ifndef vector_hpp
#define vector_hpp

// ...

template <typename T>
void vector<T>::resize(int newsize)
{
// ...
}

// ...

#endif  // vector_hpp
```

notare la ripetizione di `template <typename T>`

definizione

Recap

- Utilizzo dei template
- File header
- Inclusione atipica del file `.hpp`