

# Polimorfismo de coerção (e typecasting)

Integrantes:

Gustavo Lopes

Lucas Santiago

Thiago Henriques

# Definição

*“A coercion is instead a semantic operation that is needed to convert an argument to the type expected by a function, in a situation that would otherwise result in a type error.”*

*“On Understanding Types, Data Abstraction, and Polymorphism”,  
page 6.*

# Como ele é feito?

No polimorfismo de coerção a compilação se encarrega de escolher a função mais abrangente\* e fazer os castings necessários para ter a tipação escolhida pelo programador.

# Quais vantagens?

- Ganho de espaço em disco
- Polimorfismo (sobrecarregar o mesmo nome de função)
- Facilitar a vida do programador

# Exemplo #1

```
float b = 2;
```

```
int a = 3.33;      // coerção implícita
```

```
int a = (int) 10.2; // coerção explícita (casting)
```

## Exemplo #2

```
// int a, b;  
int soma(int a, int b){  
    return a + b;  
}
```

```
int resultado = soma(a, b);
```

## Exemplo #3

```
// float a, b;  
float soma(float a, float b){  
    return a + b;  
}
```

  

```
float resultado = soma(a, b);
```

## Exemplo #4

```
// int a, b;  
float soma(float a, float b){  
    return a + b;  
}
```

```
int resultado = (int) soma(a, b); // casting
```



## Exemplo #5

```
// int a, b;  
float soma(float a, float b){  
    return a + b;  
}
```

```
int resultado = soma(a, b); // coerção
```

## Exemplo #5

```
int resultado = soma(a, b); // coerção
```

A compilação deste código adiciona os castings necessários implicitamente