

An abstract, light blue geometric pattern consisting of overlapping circles and lines, resembling a complex network or a stylized molecular structure, is positioned in the upper left corner of the slide.

# Contratos e Exceções

# Papéis do programador

- Relativamente a um módulo o programador pode ser
  - Produtor – Se o desenvolveu total ou parcialmente
  - Consumidor – Se o usa de alguma forma

# Contratos: documentação e verificação

```
/**
 * Returns the double closest to the square root of its argument
 * value.
 *
 * @pre 0 ≤ value
 * @post ...
 *
 * @param value the value whose square root will be approximated
 * @throws IllegalArgumentException If value < 0
 */
static public double squareRoot(final double value) {
    if(value < 0)
        throw new IllegalArgumentException("Message");
    ...
    assert ... : "Informative message";
    return result;
}
```

As asserções, por omissão, estão desactivadas. Para que tenham algum efeito tem de se executar a JVM com a opção -ea.

# Invariante de instância

```
public Rational(final int numerator,
                final int denominator) {
    if(denominator == 0)
        throw new IllegalArgumentException("Message");
    this.numerator = numerator;
    this.denominator = denominator;
    normalize();
    checkInvariant();
}
```

Onde? No final dos construtores, no início de métodos não modificadores e no início e no fim de métodos modificadores.

```
private void checkInvariant() {
    assert 0 < denominator : "...";
    assert gcd(numerator, denominator) == 1 : "...";
}
```

O invariante de instância é a condição que todas as instâncias de uma classe têm de verificar para que estejam num estado correcto.

# Métodos privados

```
private void reduce() {  
    assert denominator != 0;  
  
    if (denominator < 0) {  
        denominator = -denominator;  
        numerator = -numerator;  
    }
```

Pré-condição de método privado  
(parte da implementação da  
classe) verificada com asserção.

```
    final int gcd = gcd(numerator, denominator);  
    numerator /= gcd;  
    denominator /= gcd;
```

Nos métodos privados não se verifica  
o invariante de instância.

```
    assert gcd(numerator, denominator) == 1 : "...";  
    assert 0 < denominator : "...";  
}
```

# Sumário

- Erros e Exceções