

Questão 01:

R: Após a execução será impresso:

```
/home/dev/.jdk/azul-17.0.6/bin/java -j
Classe A
Classe B

Process finished with exit code 0
```

Ele apresenta esse resultado pois a classe herda de A e, sendo chamado o construtor da classe A primeiro imprimindo “Classe A” e em sequência sendo chamado o construtor da classe B, sendo impresso a frase “Classe B” depois é definido a classe C.

Questão 06:

R: Para a sobrescrita de um método primeiro ela deve estar criada na classe pai por exemplo:

```
2 usages 1 override new *
public double getSalarioPrimeiraParcela() { return this.salario * 0.6; }

2 usages 1 override new *
public double getSalarioSegundaParcela() { return this.salario * 0.4; }
```

onde os métodos `getSalarioPrimeiraParcela` e `getSalarioSegundaParcela` pertencem a classe `Funcionario`, e para ser sobrescrita na classe `Professor` ele deve estender a classe `Funcionario` e deve escrever o novo método com o mesmo nome da classe pai

```
@Override
public double getSalarioPrimeiraParcela() { return this.getSalario(); }

2 usages new *
@Override
public double getSalarioSegundaParcela() { return 0.0; }
```

no exemplo acima os métodos da classe pai retornava 60% do valor do salário no primeiro método e 40% no segundo método, e na classe filha retorna 100% do valor do salário e 0% do valor no segundo método.

Questão 07:

R: Na sobrecarga os métodos devem estar na mesma classe, e ter o mesmo nome, um bom exemplo é o construtor, que tem a possibilidade de criar o objeto com vários parâmetros ou com nenhum parâmetro (construtor vazio), segue exemplo abaixo:

```
public Pessoa() {
}

2 usages new *
public Pessoa(String nome, String sobrenome) {
    this.nome = nome;
    this.sobrenome = sobrenome;
}
```