



# Desenvolvimento de Sistemas

## Centro Universitário UNA

Leo Larback – [leonardo.cabral@prof.una.br](mailto:leonardo.cabral@prof.una.br) – <http://www.larback.com.br>

### Exercício – Revisão Geral

1. Dado o seguinte código:

```
interface Simples {  
    boolean mostra( );  
    byte ver(short s);  
}
```

Qual fragmento de código irá compilar? (Marque todas as corretas)

- A. `interface Simples2 implements Simples { }`
- B. `abstract class Class2 extends Simples {  
 public boolean mostra( ) { return true; } }`
- C. `abstract class Class2 implements Simples { }`
- D. `abstract class Class2 implements Simples {  
 public boolean mostra( ) { return (true) ; } }`
- E. `class Class2 implements Simples {  
 boolean mostra( ) {return false; }  
 byte ver(short s) { return 42; } }`

2. Qual das seguintes opções declara uma classe abstract compilável? (Marque todas as Corretas.)

- A. `public abstract class Canine { public String speak( ); }`
- B. `public abstract class Canine { public String speak ( ) { } }`
- C. `public class Canine { public abstract String speak ( ); }`
- D. `public class Canine abstract { public abstract String speak ( ); }`

3. Qual afirmativa está correta?

- A. "X estende Y" é correto se, e somente se, X for uma classe e Y for uma interface.
- B. "X estende Y" é correto se, e somente se, X for uma interface e Y for uma classe.
- C. "X estende Y" é correto se X e Y forem ambas classes ou ambas interfaces.
- D. "X estende Y" é correto para todas as combinações de X e Y sendo classes e/ou interfaces.

4. Quais das seguintes são declarações válidas? (Marque todas as corretas.)

- A. `int $x;`
- B. `int 123;`
- C. `int _123;`
- D. `int #dim;`
- E. `int *divide;`
- F. `int central_sales_region_Summer_200S_gross_sa1es`

5. Quais das seguintes opções são declarações válidas? (Marque todas as corretas.)

- A. `short x [ ];`
- B. `short [ ] y;`
- C. `short [5] x2;`
- D. `short z2 [5] ;`
- E. `short [ ] z [ ] [ ] ;`
- F. `short [ ] y2 = [5] ;`

6. Quais afirmativas são verdadeiras? (Marque todas as corretas)

- A. Os relacionamentos Tem-Um sempre dependem da herança.
- B. Os relacionamentos Tem-Um sempre dependem das variáveis de instâncias.
- C. Os relacionamentos Tem-Um sempre precisam de pelo menos dois tipos de classes.

D. Os relacionamento Tem-Um sempre dependem do polimorfismo.

7. Usando os fragmentos abaixo, complete o seguinte código de forma que ele compile. Observação, talvez não seja preciso preencher todos os espaços em branco.

Código:

```
class Agedp{
    _____
    public Agedp(int x) {
        _____
    }
}

public class Kinder extends Agedp {
    _____
    public Kinder (int x) {
        _____ ();
    }
}
```

Fragmentos: Use cada um dos seguintes fragmentos quantas vezes for preciso, ou deixe sem uso:

Agedp	super	this	
(	)	{	}
;			

8. Enumere as 2 e 3 colunas de acordo com a primeira:

(1) boolean	( ) Depende da JVM	( ) lógico
(2) char	( ) 8 bits	( , , , ) inteiro
(3) int	( , ) 16 bits	( , ) real
(4) byte	( , ) 32 bits	( ) caracter
(5) double	( , ) 64 bits	
(6) long		
(7) short		
(8) float		

9. Dado o seguinte código:

```
public class Carro {
    private int cavalos;
    private String combustivel;
    public Carro(int c) {
        this.cavalos = c;
    }
    public Carro(String combustivel) {
        this.combustivel = combustivel;
    }
}
```

Marque as opções que criam corretamente objetos de Carro

- a) Carro car = new Carro();
- b) Carro car = new Carro;
- c) Carro car = new Carro(100);
- d) Carro car = new Carro("Gasolina");

10. Analise o código abaixo e responda o que se pede:

```
public class Follow {
    public Follow () {
        System.out.println("0")
    }
    public Follow (int a) {
        this ();
        System.out.println("1");
    }
    public Follow (int a, int b) {
        this (0);
        System.out.println("2");
    }
    public Follow (int a, int b, int c) {
        this (0,0);
        System.out.println("3");
    }
    public Follow (int a, int b, int c, int d) {
        this ();
        System.out.println("4");
    }
}
```

O que seria impresso ao executarmos:

- a) new Follow();
  
- b) new Follow(10,10,10);
  
- c) new Follow (10);
  
- d) new Follow (10,10);