

Exercícios Unidade00i:

1)

- Faça o quadro de memória do programa abaixo

```
class Ponteiro02PassagemTipoPrimitivo {  
    public static void passagemDeTipoPrimitivo(int a){  
        escrever("a: " + a);  
        a = 10;  
        escrever("a: " + a);  
    }  
    public static void main(String[] args) {  
        int x = 5;  
        escrever("x: " + x);  
        passagemDeTipoPrimitivo(x);  
        escrever("x: " + x);  
    }  
}
```

Resposta:

Memoria	
x: 5	75h
a: 5 => 10	76h
	78h
	7Ah

Tela	
x: 5	
a: 5	
a: 10	
x: 5	

2)

- Faça o quadro de memória e mostre a saída na tela

```
class Ponteiro03PassagemArray {
    public static void passagemDeArray(int[] b){
        for (int i = 0; i < 5; i++){
            b[i] *= 5;    escrever("b[" + i + "]: " + b[i]);
        }
        b = new int [5];
        for (int i = 0; i < 5; i++){
            b[i] = i;    escrever("b[" + i + "]: " + b[i]);
        }
    }
    public static void main(String[] args) {
        int [] y = new int [5];
        for (int i = 0; i < 5; i++){
            y[i] = i;    escrever("y[" + i + "]: " + y[i]);
        }
        passagemDeArray(y);
        for (int i = 0; i < 5; i++){
            escrever("y[" + i + "]: " + y[i]);
        }
    }
}
```

Resposta:

Memoria	
Y: 7Eh	75h
B: 7Eh => 78h	76h
	77h
0	78h
1	79h
2	7Ah
3	7Bh
4	7Ch
	7Dh
0	7Eh
1 => 5	7Fh
2 => 10	80h
3 => 15	81h
4 => 20	82h

Tela
Y[0]: 0
Y[1]: 1
Y[2]: 2
Y[3]: 3
Y[4]: 4
B[0]: 0
B[1]: 5
B[2]: 10
B[3]: 15
B[4]: 20
B[0]: 0
B[1]: 1
B[2]: 2
B[3]: 3
B[4]: 4
y[0]: 0
y[1]: 5
y[2]: 10
y[3]: 15
y[4]: 20

3)

• Faça o quadro de memória do programa abaixo

```
class Ponteiro04Objeto {
    public static void main (String[] args){
        Cliente c1 = null, c2 = null, c3 = null;
        escrever("ADDRs:\nc1(" + c1 + ")\nc2(" + c2 + ")\nc3(" + c3 + ")");
        c1 = new Cliente(1, "aa");      c2 = c1;    c3 = new Cliente(2, "bb");
        escrever("ADDRs:\nc1(" + c1 + ")\nc2(" + c2 + ")\nc3(" + c3 + ")");
        c2.setCodigo(3);
        escrever("ATRIBUTOs:");
        escrever("c1(" + c1.getCodigo() + " / " + c1.getNome() + ")");
        escrever("c2(" + c2.getCodigo() + " / " + c2.getNome() + ")");
        escrever("c3(" + c3.getCodigo() + " / " + c3.getNome() + ")");
    }
}
```

Resposta:

Memoria			
C1: null => 79h			75h
C2: null => 79h			76h
C3: null => 7Ah			77h
			78h
1 => 3	"aa"		79h
2	"bb"		7Ah
			7Bh
			7Ch
			7Dh
			7Eh
			7Fh
			80h

Tela
ADDRs:
C1: null
C2: null
C3: null
ADDRS:
C1: 79h
C2: 79h
C3: 7Ah
ATRIBUTOs:
C1(3 / aa)
C2(3 / aa)
C3(2 / bb)

4)

• Faça o quadro de memória do programa abaixo

```
class Ponteiro05PassagemObjeto {
    public static Cliente setar2(Cliente y){
        y.setCodigo(6); y.setNome("ff");
        return y;
    }
    public static void setar1(Cliente x){
        x.setCodigo(4); x.setNome("dd"); x = new Cliente (5, "ee");
    }
    public static void main (String[] args){
        Cliente c1 = new Cliente(1, "aa"), c2 = null; c3 = new Cliente(2, "bb");
        c2 = c1;
        setar1(c1);
        c3 = setar2(c2);
    }
}
```

Resposta:

Memoria				
C1: 79h				75h
C2: null => 79h				76h
C3: 7Ah => 79h				77h
				78h
1	aa			79h
4	dd			
6	ff			
2	bb			7Ah
5	ee			7Bh
				7Ch
X: 79h => 78h				7Dh
Y: 79h				7Eh
				7Fh
				80h

5)

- Na verdade, no comando `c2 = c1` do exercício anterior, o programador gostaria que os atributos do objeto apontado por `c2` fossem iguais aos do objeto apontado por `c1`, contudo, apontando para objetos distintos. Como podemos ajudá-lo?

Resposta:

O programador poderia utilizar um método para copiar o conteúdo de `c1` para `c2`

```
public static Cliente copia(){  
    Cliente copia = new Cliete();  
    copia.codigo = this.codigo;  
    copia.nome = this.nome;  
    return copia;  
}
```

6)

- Mostre a alteração anterior na classe Ponteiro04Objeto

```
class Ponteiro07ObjetoClone {
    public static void main (String[] args){
        Cliente c1 = null, c2 = null, c3 = null;
        escrever("ADDRs:\nc1(" + c1 + ")\nc2(" + c2 + ")\nc3(" + c3 + ")");
        c1 = new Cliente(1, "aa"); c2 = c1.clone(); c3 = new Cliente(2, "bb");
        escrever("ADDRs:\nc1(" + c1 + ")\nc2(" + c2 + ")\nc3(" + c3 + ")");
        c2.setCodigo(3);
        escrever("ATRIBUTOs:");
        escrever("c1(" + c1.getCodigo() + " / " + c1.getNome() + ")");
        escrever("c2(" + c2.getCodigo() + " / " + c2.getNome() + ")");
        escrever("c3(" + c3.getCodigo() + " / " + c3.getNome() + ")");
    }
}
```

Resposta:

Memoria			
C1: null => 79h			75h
C2: null => 7Bh			76h
C3: null => 7Ah			77h
			78h
1 => 3	"aa"		79h
2	"bb"		7Ah
1 => 3	"aa"		7Bh
			7Ch
			7Dh
			7Eh
			7Fh
			80h

Tela
ADDRs:
C1: null
C2: null
C3: null
ADDRs:
C1: 79h
C2: 79h
C3: 7Ah
ATRIBUTOs:
C1(3 / aa)
C2(3 / aa)
C3(2 / bb)

7)

- Um estudante de Algoritmos e Estruturas de Dados (em JAVA) implementou uma classe Hora, cujo construtor recebe e armazena uma hora, minuto e segundo. O que acontece se a classe X abaixo for colocada na mesma pasta que a classe Hora?

```
class X {  
    public static void main (String[] args){  
        Hora h1 = new Hora(12, 30, 30);  
        Hora h2 = new Hora(12, 30, 30);  
        if (h1 == h2)  
            System.out.println("Identicos!");  
        else  
            System.out.println("Diferentes!");  
    }  
}
```

- A) Escreve na tela "Identicos!".
- B) Escreve na tela "Diferentes".
- C) Erro de compilação.
- D) Erro de execução na linha do if.
- E) Erro de execução na declaração objetos.

Resposta:

- B) Escreve na tela "Diferentes".

8)

Seja a classe X abaixo e a Animal implementada e não mostrada, avalie as afirmações listadas a seguir.

```
class X {  
    public static void main (String[] args){  
        Animal a = new Animal ("Cao", 32, 'a');  
        Animal b = new Animal ("Cao", 'x');  
        Animal c = b;  
        c.nome = "Gato";  
        System.out.println(b.nome);  
        c.setIdade(45);  
    } }  
}
```

I – Possivelmente, a Classe Animal tem três ou mais atributos. Além disso, no construtor com três parâmetros, o atributo que recebe valor do primeiro parâmetro pode ser do tipo String e os que recebem os outros dois podem ser do tipo int.

II - O comando System.out.println(b.nome) imprime a palavra "Gato".

III - A classe Animal deve ter um atributo idade e esse será obrigatoriamente privado.

IV - Na classe animal o atributo nome tem que ser estático.

É correto apenas o que se afirma em: A) I e II. B) II e III. C) III e IV. D) I, II e III.

Resposta:

- A) I e II.