

- O que significa cada um dos trechos de código abaixo?

```
int [ ] vet
```

```
= new int [5]
```

```
int [ ] vet = new int [5];
```

int[] vet: define um vetor sem espaço (ponteiro) definido e o nome;

= new int [5]: reserva 5 espaços de inteiro em um vetor;

int[] vet = new int[5]: define o nome do vetor e reserva um espaço na memória de 5 inteiros;

- Explique o que o programa abaixo imprime na tela

```
class Ponteiro01Array {  
  
    public static void main (String[] args) {  
        int[] vet = new int [5];  
        escrever(vet);  
  
        vet = new int [5];  
        escrever(vet);  
    }  
}
```

O programa imprime o endereço do valor inteiro da posição 0 do array vet e após redefinir o vetor (“criar” um novo), ele imprime novamente o primeiro valor.

- 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);
    }
}
```

Memória	Tela
x = 5;	
x = 5;	x: 5
x = 5;	a: 5
x= 5; a = 10	
x= 5; a = 10	a: 10
x = 5	
x = 5	x: 5

- 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]);
        }
    }
}
```

	Memória	Tela
75h	y = 0 // b = 0	y[0]: 0
76h	y = 1 // b = 5	y[1]: 1
77h	y = 2 // b = 10	y[2]: 2
78h	y = 3 // b = 15	y[3]: 3
7Ah	y = 4 // b = 20	y[4]: 4
7Bh	b = 0	b[0]: 0
7Ch	b = 1	b[1]: 5
7Dh	b = 2	b[2]: 10
7Eh	b = 3	b[3]: 15
7Fh	b = 4	b[4]: 20
80h		b[0]: 0
81h		b[1]: 1
82h		b[2]: 2

83h		b[3]: 3
		b[4]: 4
		y[0]: 0
		y[1]: 5
		y[2]: 10
		y[3]: 15
		y[4]: 20

• 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() + ")");
    }
}

```

Memória	Tela
c1 = null // c1 = 1, "aa" // c1 = 3, "aa"	ADDRs: c1(null)
c2 = null // c2 = c1 // c2 = 3, "aa"	c2(null)
c3 = null // c3 = 2, "bb"	c3(null)
	ADDRs: c1(7Ah)
	c2(7Ah)
	c3(A5h)
	ATRIBUTOs:

	c1(1 / "aa")
	c2(3 / "aa")
	c3(2 / "bb")

- 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);
    } }

```

Memória
c1 = 1, "aa" // 4, "dd" // 5, "ee" // 6, "ff"
c2 = null // 1, "aa" // 4, "dd" // 5, "ee" // 6, "ff"
c3 = 2, "bb" / 6, "ff"

• Mostre a alteração anterior na classe Ponteiro04Objeto

```
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() + ")");
    }
}
```

Memória	Tela
c1 = null // 1, aa	ADDRs:
c2 = null // 1, aa // 3, aa	c1(null)
c3 = null // 2, bb	c2(null)
	c3(null)
	ADDRs:
	c1(7Ah)
	c2(9Ah)
	c3(A5h)
	ATRIBUTOs
	c1(1 / aa)
	c2(3 / aa)
	c3(2 / bb)

- Mostre o quadro de memória para o programa abaixo

```
class Ponteiro08Objeto {  
    public static void main (String[] args){  
        Cliente c1 = new Cliente(1, "aa");  
        Cliente vet[] = new Cliente [5];  
        sop(c1 + "/" + c1.getCodigo() + "/" + c1.getNome());  
        for (int i = 0; i < vet.length; i++){  
            vet[i] = c1.clone();  
            System.out.println(vet[i] + "/" + vet[i].getCodigo() + "/" + vet[i].getNome());  
        }  
    }  
}
```

Memória
c1 = 1, aa
vet[0] = 1, aa
vet[1] = 1, aa
vet[2] = 1, aa
vet[3] = 1, aa
vet[4] = 1, aa

- 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.

b) Escreve na tela "Diferentes" - porque o h1 e h2 estão se referindo ao endereço de memória do objeto Hora, e são dois endereços diferentes.

- 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.

a) I e II