

Exercícios

Tente resolver manualmente e, em seguida, programe o código para descobrir as respostas

Exercício: determine se cada um desses arquivos será compilado e executado sem erros. Se não for, como você os corrigiria?

A

```
class Books {
    String title;
    String author;
}

class BooksTestDrive {
    public static void main(String[] args) {
        Books[] myBooks = new Books[3];
        int x = 0;
        myBooks[0].title = "The Grapes of Java";
        myBooks[1].title = "The Java Gatsby";
        myBooks[2].title = "The Java Cookbook";
        myBooks[0].author = "bob";
        myBooks[1].author = "sue";
        myBooks[2].author = "ian";

        while (x < 3) {
            System.out.print(myBooks[x].title);
            System.out.print(" by ");
            System.out.println(myBooks[x].author);
            x = x + 1;
        }
    }
}
```

B

```
class Hobbits {
    String name;

    public static void main(String[] args) {
        Hobbits[] h = new Hobbits[3];
        int z = 0;

        while (z < 4) {
            z = z + 1;
            h[z] = new Hobbits();
            h[z].name = "bilbo";
            if (z == 1) {
                h[z].name = "frodo";
            }
            if (z == 2) {
                h[z].name = "sam";
            }
            System.out.print(h[z].name + " is a ");
            System.out.println("good Hobbit name");
        }
    }
}
```

Exercício: reconstrua os trechos de código para criar um programa Java funcional que produza o resultado listado abaixo

```
int y = 0;
```

```
ref = index[y];
```

```
islands[0] = "Bermuda";  
islands[1] = "Fiji";  
islands[2] = "Azores";  
islands[3] = "Cozumel";
```

```
int ref;  
while (y < 4) {
```

```
System.out.println(islands[ref]);
```

```
index[0] = 1;  
index[1] = 3;  
index[2] = 0;  
index[3] = 2;
```

```
String [] islands = new String[4];
```

```
System.out.print("island = ");
```

```
int [] index = new int[4];
```

```
y = y + 1;
```

```
class TestArrays {  
  
    public static void main(String [] args) {
```

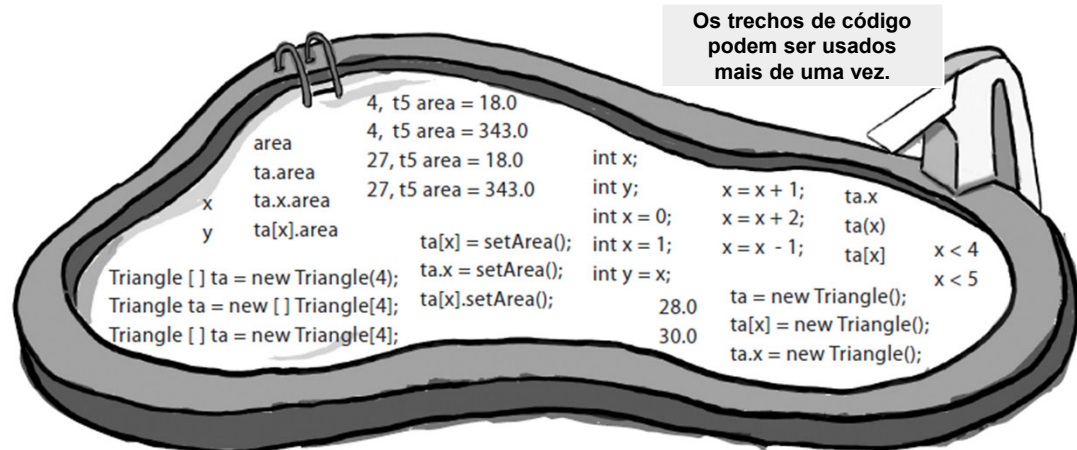
Resultado:

```
File Edit Window Help Sunscreen  
% java TestArrays  
island = Fiji  
island = Cozumel  
island = Bermuda  
island = Azores
```

Exercício: pegue trechos de código da piscina e coloque-os corretamente nas linhas em branco do código para produzir o resultado abaixo

```
class Triangle {
    double area;
    int height;
    int length;

    public static void main(String[] args) {
        _____
        _____
        while ( _____ ) {
            _____
            _____
            _____
            _____
            System.out.print("triangle " + x + ", area");
            System.out.println(" = " + _____ .area);
            _____
        }
        _____
        x = 27;
        Triangle t5 = ta[2];
        ta[2].area = 343;
        System.out.print("y = " + y);
        System.out.println(", t5 area = " + t5.area);
    }
    void setArea() {
        _____ = (height * length) / 2;
    }
}
```



Resultado:

```
File Edit Window Help Bermuda
%java Triangle
triangle 0, area = 4.0
triangle 1, area = 10.0
triangle 2, area = 18.0
triangle 3, area = ____
y = _____
```

Bonus Question!

For extra bonus points, use snippets from the pool to fill in the missing output (above).

Exercício: determine quais variáveis de referência se referem a quais objetos.

```
class HeapQuiz {
    int id = 0;

    public static void main(String[] args) {
        int x = 0;
        HeapQuiz[] hq = new HeapQuiz[5];
        while (x < 3) {
            hq[x] = new HeapQuiz();
            hq[x].id = x;
            x = x + 1;
        }
        hq[3] = hq[1];
        hq[4] = hq[1];
        hq[3] = null;
        hq[4] = hq[0];
        hq[0] = hq[3];
        hq[3] = hq[2];
        hq[2] = hq[0];
        // do stuff
    }
}
```

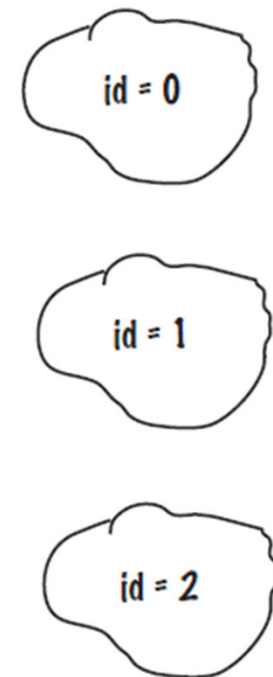
**Cada variável de referência
com seu(s) respectivo(s)
objeto(s).**

**Você pode não precisar usar
todas as referências.**

Reference Variables:



HeapQuiz Objects:



Exercício: explique a diferença e por quê a primeira implementação é uma solução melhor para atualizar a lista de contatos de um telefone.

1.

```
Contact [] contacts = new Contact[10];
while (x < 10 ) {    // make 10 contact objects
    contacts[x] = new Contact();
    x = x + 1;
}
// do complicated Contact list updating with contacts
```

2.

```
Contact contactRef;
while ( x < 10 ) {    // make 10 contact objects
    contactRef = new Contact();
    x = x + 1;
}
// do complicated Contact list updating with contactRef
```