

Exercícios

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

Exercício: considerando o método abaixo, quais das chamadas de método listadas à direita são corretas?

```
int calcArea(int height, int width) {  
    return height * width;  
}
```

a) `int a = calcArea(7, 12);`
`short c = 7;`
`calcArea(c, 15);`

b) `int d = calcArea(57);`

c) `calcArea(2, 3);`

d) `long t = 42;`
`int f = calcArea(t, 17);`

e) `int g = calcArea();`

f) `calcArea();`

g) `byte h = calcArea(4, 20);`

h) `int j = calcArea(2, 3, 5);`

Exercício: determine se cada um dos arquivos abaixo é compilável e executado sem erros. Se não for o caso, como você os corrigiria?

A

```
class XCopy {

    public static void main(String[] args) {
        int orig = 42;
        XCopy x = new XCopy();
        int y = x.go(orig);
        System.out.println(orig + " " + y);
    }

    int go(int arg) {
        arg = arg * 2;
        return arg;
    }
}
```

B

```
class Clock {
    String time;

    void setTime(String t) {
        time = t;
    }

    void getTime() {
        return time;
    }
}

class ClockTestDrive {
    public static void main(String[] args) {
        Clock c = new Clock();

        c.setTime("1245");
        String tod = c.getTime();
        System.out.println("time: "+tod);
    }
}
```

Exercício: faça a correspondência entre os blocos de código candidatos e o resultado que você veria se os blocos fossem inseridos.

```
public class Mix4 {
    int counter = 0;

    public static void main(String[] args) {
        int count = 0;
        Mix4[] mixes = new Mix4[20];
        int i = 0;
        while (  ) {
            mixes[i] = new Mix4();
            mixes[i].counter = mixes[i].counter + 1;
            count = count + 1;
            count = count + mixes[i].maybeNew(i);
            i = i + 1;
        }
        System.out.println(count + " " +
                           mixes[1].counter);
    }

    public int maybeNew(int index) {
        if (  ) {
            Mix4 mix = new Mix4();
            mix.counter = mix.counter + 1;
            return 1;
        }
        return 0;
    }
}
```

Candidatos

`i < 9`

`index < 5`

`i < 20`

`index < 5`

`i < 7`

`index < 7`

`i < 19`

`index < 1`

Possível resultado

14 7

9 5

19 1

14 1

25 1

7 7

20 1

20 5

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

```
public class Puzzle4 {
    public static void main(String [] args) {

        _____
        int number = 1;
        int i = 0;
        while (i < 6) {

            _____
            _____
            number = number * 10;
        }

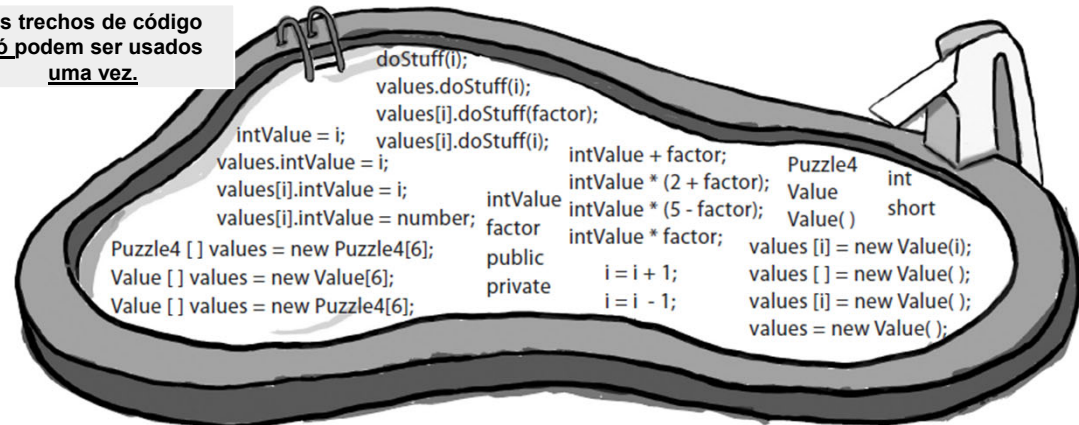
        int result = 0;
        i = 6;
        while (i > 0) {

            _____
            result = result + _____
        }
        System.out.println("result " + result);
    }
}

class _____ {
    int intValue;

    _____ doStuff(int _____) {
        if (intValue > 100) {
            return _____
        } else {
            return _____
        }
    }
}
```

Os trechos de código
só podem ser usados
uma vez.



Resultado:

```
File Edit Window Help BellyFlop
%java Puzzle4
result 543345
```