

Exercícios

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

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

```
if (x == 1) {  
    System.out.print("d");  
    x = x - 1;  
}
```

```
if (x == 2) {  
    System.out.print("b c");  
}
```

```
class Shuffle1 {  
    public static void main(String [] args) {
```

```
if (x > 2) {  
    System.out.print("a");  
}
```

```
int x = 3;
```

```
x = x - 1;  
System.out.print("-");
```

```
while (x > 0) {
```

Resultado:

```
File Edit Window Help Sleep  
% java Shuffle1  
a-b c-d
```

Exercício: determine se cada um dos arquivos é compilável (se rodam/funcionam). Caso não compilem, como você os corrigiria?

A

```
class Exercisela {  
    public static void main(String[] args) {  
        int x = 1;  
        while (x < 10) {  
            if (x > 3) {  
                System.out.println("big x");  
            }  
        }  
    }  
}
```


B

```
public static void main(String [] args) {  
    int x = 5;  
    while ( x > 1 ) {  
        x = x - 1;  
        if ( x < 3) {  
            System.out.println("small x");  
        }  
    }  
}
```

C

```
class Exerciselc {  
    int x = 5;  
    while (x > 1) {  
        x = x - 1;  
        if (x < 3) {  
            System.out.println("small x");  
        }  
    }  
}
```

Exercício: faça a correspondência entre o bloco de código candidato (à esquerda) e o resultado (à direita).

```
class Test {  
    public static void main(String [] args) {  
        int x = 0;  
        int y = 0;  
        while (x < 5) {  
              
            System.out.print(x + " " + y + " ");  
            x = x + 1;  
        }  
    }  
}
```

Candidatos:

y = x - y;

y = y + x;

y = y + 2;
if(y > 4) {
 y = y - 1;
}

x = x + 1;
y = y + x;

if (y < 5) {
 x = x + 1;
 if (y < 3) {
 x = x - 1;
 }
}
y = y + 2;

Possíveis resultados:

22 46

11 34 59

02 14 26 38

02 14 36 48

00 11 21 32 42

11 21 32 42 53

00 11 23 36 410

02 14 25 36 47

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

```
class PoolPuzzleOne {
    public static void main(String [] args) {
        int x = 0;

        while ( _____ ) {

            _____
            if ( x < 1 ) {
                _____
            }
            _____

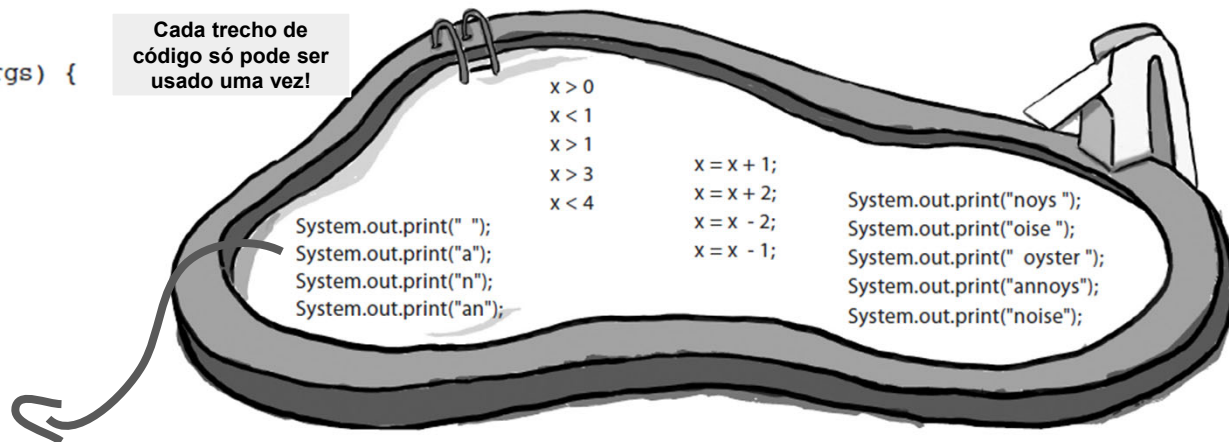
            if ( _____ ) {
                _____

                _____
            }
            if ( x == 1 ) {
                _____
            }
            if ( _____ ) {
                _____

                _____
            }
            System.out.println();

            _____
        }
    }
}
```

Cada trecho de código só pode ser usado uma vez!



Resultado:

```
File Edit Window Help Cheat
%java PoolPuzzleOne
a noise
annoys
an oyster
```

Exercício: determine se cada um dos arquivos é compilável (se rodam/funcionam). Caso não compilem, como você os corrigiria?

A

```
class StreamingSong {  
    String title;  
    String artist;  
    int duration;  
  
    void play() {  
        System.out.println("Playing song");  
    }  
  
    void printDetails() {  
        System.out.println("This is " + title +  
                           " by " + artist);  
    }  
}  
  
class StreamingSongTestDrive {  
    public static void main(String[] args) {  
  
        song.artist = "The Beatles";  
        song.title = "Come Together";  
        song.play();  
        song.printDetails();  
    }  
}
```

B

```
class Episode {  
  
    int seriesNumber;  
    int episodeNumber;  
  
    void skipIntro() {  
        System.out.println("Skipping intro...");  
    }  
  
    void skipToNext() {  
        System.out.println("Loading next episode...");  
    }  
}  
  
class EpisodeTestDrive {  
    public static void main(String[] args) {  
  
        Episode episode = new Episode();  
        episode.seriesNumber = 4;  
        episode.play();  
        episode.skipIntro();  
    }  
}
```

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

```
d.playSnare();
```

```
DrumKit d = new DrumKit();
```

```
boolean topHat = true;  
boolean snare = true;
```

```
void playSnare() {  
    System.out.println("bang bang ba-bang");  
}
```

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

```
if (d.snare == true) {  
    d.playSnare();  
}
```

```
d.snare = false;
```

```
class DrumKitTestDrive {
```

```
d.playTopHat();
```

```
class DrumKit {
```

```
void playTopHat () {  
    System.out.println("ding ding da-ding");  
}
```

```
File Edit Window Help Dance
```

```
% java DrumKitTestDrive  
bang bang ba-bang  
ding ding da-ding
```

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

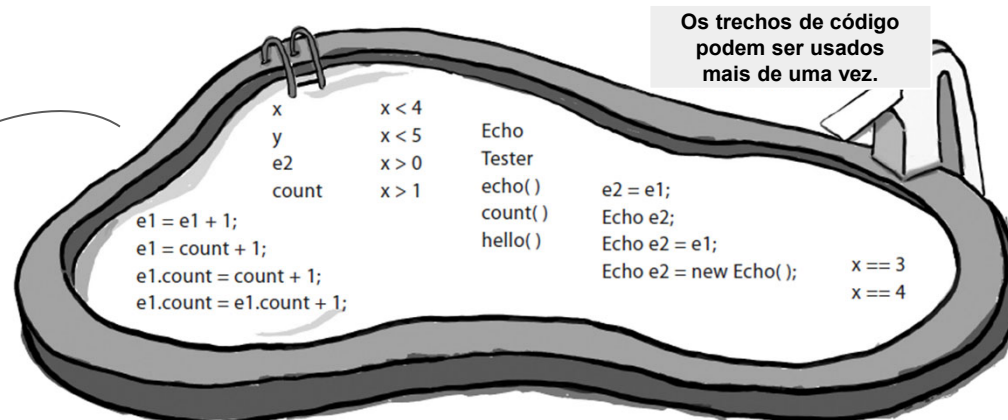
```
public class EchoTestDrive {
    public static void main(String []
args) {
        Echo e1 = new Echo();

        _____

        int x = 0;
        while ( _____ ) {
            e1.hello();

            _____

            if ( _____ ) {
                e2.count = e2.count + 1;
            }
            if ( _____ ) {
                e2.count = e2.count + e1.count;
            }
            x = x + 1;
        }
        System.out.println(e2.count);
    }
}
```



Resultado:

```
File Edit Window Help Implode
%java EchoTestDrive
helloooo...
helloooo...
helloooo...
helloooo...
10
```

```
class _____ {
    int _____ = 0;
    void _____ {
        System.out.println("helloooo... ");
    }
}
```