

Fundamentos de Java

3-10: Loops, Variáveis e Matrizes

Atividades Práticas

Objetivos da Lição:

- Criar um loop while em um construtor para criar um mundo
- Descrever um loop infinito e como impedir sua ocorrência
- Usar uma matriz para armazenar diversas variáveis usadas para criar um mundo
- Criar uma expressão com operadores lógicos
- Descrever o escopo de uma variável local em um método
- Usar variáveis de string para armazenar e concatenar strings

Vocabulário:

Identifique a palavra do vocabulário para cada definição a seguir.

	Tipo de item armazenado em uma matriz, como uma string ou um valor inteiro, acessados com um índice.
	Instrução que pode executar uma seção de código várias vezes.
	Símbolos que podem ser usados para combinar várias expressões booleanas em uma.
	Número de posição no objeto de matriz que especifica qual elemento da matriz acessar.
	Objeto que contém diversas variáveis. É possível usar um índice para acessar as variáveis.
	Variável declarada dentro do corpo do método para armazenar valores temporariamente, como referências a objetos ou inteiros.
	Loop que faz com que o código continue sendo executado. O código não para porque o final dele não foi estabelecido.

Tente/solucione:

1. Verdadeiro ou falso: Loop é uma instrução que pode executar uma seção de código uma vez. É uma forma eficiente de transmitir informações para uma instância única.

2. Qual dos seguintes componentes de um loop while não é necessário?

- a. Palavra-chave while Java
- b. Condição
- c. Uma ou mais instruções
- d. Instância

3. No exemplo de loop while a seguir, descreva o que o corpo do loop executa:

```
int i = 0;

while (i < 10,000)

{

    addObject (new Bee (), 110, 130);

    i = i + 1;

}
```

4. No exemplo de loop a seguir, descreva por que se trata de um loop infinito:

```
int i = 0;

while (i < 10,000)

{

    addObject (new Bee (), 180, 120);

}
```

5. Como você resolve um loop infinito?

6. Verdadeiro ou falso: Operadores lógicos são usados para conectar uma ou mais expressões booleanas.

7. No exemplo de código a seguir, identifique os operadores lógicos marcando-os com um círculo e defina como eles são usados:

```
public void act()
{
    if (!isDown && Greenfoot.isKeyDown ("x")) {
        setImage ("dog.png");
        isDown = true;
    }

    if (isDown && !Greenfoot.isKeyDown("y")){
        setImage ("dog2.png");
        isDown = false;
    }
}
```

8. Escreva a instrução que acessa a chave "a" na seguinte matriz:

```
private String[] keyboardNames = {"a", "b", "c", "d", "e", "f", "g", "h"};
```

9. Em que a variável de local abaixo é inicializada?

```
int i = 5;
```

Nota: O resto das perguntas se refere ao projeto Barril.

10. Crie uma variável de campo no Avião denominada isGoingFast que armazenará um valor booleano.
11. Modifique o método handleMovement() para que a variável isGoingFast seja definida como true quando a seta para cima for clicada e definida como false se não for.
12. Adicione código ao método handleMovement() de modo que se a seta para cima for pressionada, a imagem mude para airplaneFaster.png.
13. Modifique o método animate() de modo que não seja animado se isGoingFast for verdadeiro;
14. Modifique o construtor BarrelWorld de modo que você use um loop para adicionar 3 Foguetes ao mundo.
15. Modifique o construtor BarrelWorld de modo que você use um loop para adicionar 3 Barris ao mundo.

16. Pergunta opcional – pense em maneiras de expandir seu programa com recursos adicionais. Eles podem incluir:

- Nuvens que flutuam cobrindo o avião, barris e foguetes se os dois ocuparem o mesmo local.
- Foguetes que mudam de direção
- Foguetes que vêm de lados diferentes
- Animar o barril afundando e flutuando
- Adicionar efeitos sonoros ao barril afundando e flutuando
- Qualquer ideia que você tenha