

Fundamentos de Java

7-1: Classes, Objetos e Métodos

Atividades Práticas

Objetivos da Lição:

- Reconhecer o formato geral correto de uma classe
- Criar um objeto de uma classe
- Criar métodos que sejam compilados sem erros
- Retornar um valor de um método
- Usar parâmetros em um método
- Criar uma classe de driver e adicionar instâncias de classes de Objeto
- Adicionar um construtor a uma classe
- Aplicar o novo operador
- Descrever coleta de lixo e finalizadores
- Aplicar a referência this
- Adicionar um construtor para inicializar um valor

Vocabulário:

Identifique a palavra do vocabulário para cada definição a seguir.

	Um modelo usado para criar objetos Java.
	Uma palavra-chave opcional usada para acessar os membros e métodos de uma classe.
	Uma instância de uma classe.
	O operador usado para criar uma instância de uma classe.
	Uma função incorporada da Java VM que libera memória à medida que os objetos não são mais necessários ou referenciados.
	Um método que altera o estado de um objeto.
	Um método que retorna informações sobre um objeto para o programa que o aciona.
	Um procedimento (altera o estado de um objeto) ou função (retorna informações sobre um objeto) que é encapsulado como parte de uma classe.
	Um verbo usado para descrever o ato de criar um objeto de classe usando a palavra-chave "new".
	O processo de atribuir um valor padrão a uma variável.
	Uma referência de objeto que não foi instanciada.
	Um método opcional que é chamado antes de um objeto ser removido pelo coletor de lixo.

	O nome de uma variável que é associada a um objeto.
	Um método especial usado para criar uma instância de uma classe.

Tente/solucione:

1. Crie uma classe Shape simples que representará uma forma bidimensional com segmentos de linha para as bordas. Ela deve ter a seguinte instância a variável: numSides (int), regular (booleano). Crie pelo menos dois construtores e métodos getter e setter.
2. Identifique as partes principais da Classe Java abaixo. Coloque asteriscos ao lado de todas as variáveis da instância. Coloque uma caixa ao redor de cada construtor. Circule a assinatura dos métodos diferentes do método do construtor. Coloque triângulos ao redor dos parâmetros. Sublinhe os tipos de métodos de retorno.

```

public class Animal {
    int weight, height;
    double speed;
    Animal() {
        weight = 50;
        height = 4;
        speed = 2; //miles per hour
    }
    Animal(int w, int h, int s ) {
        weight = w;
        h = height;
        speed = s
    }
    public double getTime(double miles) { //gets the number of hours to go these miles
        return miles/speed;
    }
    public int getWeight() {
        return weight;
    }
    public int getHeight() {
        return height;
    }
    public double getSpeed() {
        return speed;
    }
}

```

3. Escreva um código para criar duas instâncias do modelo de classe Animal listado no problema 2. Certifique-se de usar cada um dos dois construtores fornecidos. Em seguida, adicione o código Java que imprimirá o seguinte:
- Animal #1 tem uma velocidade de ____.
 - Animal #2 tem uma velocidade de ____.

Verifique se os espaços em branco são automaticamente preenchidos com as velocidades reais. Use os métodos fornecidos para acessar as velocidades.

4. Escreva uma classe Student. Ela deve ter as seguintes variáveis da instância para o nome, créditos, média geral e pontos de qualidade. Crie um método construtor. Crie dois métodos da seguinte forma:
- Um método que retornará a média atual que será os pontos de qualidade divididos pelos créditos.
 - Um método que considerará os créditos de uma classe ou semestre com os pontos de qualidade. Ele deve atualizar os créditos, os pontos de qualidade e a média geral.

5. Usando a classe que você criou na etapa 4, crie essas instâncias da Classe Student na tabela abaixo:

Name	Credits	Quality Points
Mary Jones	14	46
John Stiner	60	173
Ari Samala	31	69

6. Usando as variáveis da instância criadas na etapa 5, adicione 13 créditos e 52 pontos de qualidade ao aluno "Ari Samala".
7. Usando a classe Card dos slides, teste o programa para ver se funciona. Adicione uma segunda Card aleatória. O código é incluído abaixo:

```
public class Card{
    String suit,name;
    int points;

    Card(int n1, int n2){
        suit = getSuit(n1);
        name = getName(n2);
        points = getPoints(name);
    }

    public String toString(){
        return "The " + name + " of " + suit;
    }

    public String getName(int i){
        if(i == 1) return "Ace";
        if(i == 2) return "Two";
        if(i == 3) return "Three";
    }
}
```

```

        if(i == 4)    return "Four";
        if(i == 5)    return "Five";
        if(i == 6)    return "Six";
        if(i == 7)    return "Seven";
        if(i == 8)    return "Eight";
        if(i == 9)    return "Nine";
        if(i == 10)   return "Ten";
        if(i == 11)   return "Jack";
        if(i == 12)   return "Queen";
        if(i == 13)   return "King";
        return "error";
    }

    public int getPoints(String n){
        if(n == "Jack" || n == "Queen" || n == "King" || n == "Ten")
            return 10;
        if(n == "Two")
            return 2;
        if(n == "Three")
            return 3;
        if(n == "Four")
            return 4;
        if(n == "Five")
            return 5;
        if(n == "Six")
            return 6;
        if(n == "Seven")
            return 7;
        if(n == "Eight")
            return 8;
        if(n == "Nine")
            return 9;
        if(n == "Ace")
            return 1;
        return -1;
    }

    public String getSuit(int i){
        if(i == 1)    return "Diamonds";
        if(i == 2)    return "Clubs";
        if(i == 3)    return "Spades";
        if(i == 4)    return "Hearts";
        return "error";
    }
}

```

```

public class Main {

    public static void main(String args[]){

        int suitNumber = (int) (Math.random()*4.0+1);
        int faceNumber = (int) (Math.random()*13.0+1);
        Card newCard = new Card(suitNumber,faceNumber);
        System.out.println(newCard);

    }

}

```

8. Adicione o código à classe Main na etapa 7 ao seguinte:

- a. Exiba o valor total dos pontos das duas cartas aleatórias.
- b. Pergunte ao usuário se ele deseja outra carta. Se ele disser que sim, exiba a nova carta e os pontos de todas as 3 cartas da "Hand" dele.
- c. Execute o loop para permitir que o usuário continue adicionando cartas à mão até o número de pontos ultrapassar 21 ou o usuário decidir não adicionar mais cartas ou o número total de cartas ser igual a 5.