Módulo 3

Métodos e String

Programação Orientada a Objetos I

Java

(Rone Ilídio)

Métodos

- São módulos de programas, ou seja, trechos de código com determinada função.
- Cada método deve possuir um nome (identificador)
- Declaração básica de um método:

```
valor_retorno nome_metodo( tipo p1, tipo p2, ... ){
    . . . seqüência de comandos . . .
}
```

```
import java.awt.Container;
import javax.swing.*;
public class SquareInteger extends JApplet {
   public void init(){
      JTextArea outputArea = new JTextArea();//JTextArea com os resultados
      Container container = getContentPane(); //Obtém o container do applet
      container.add(outputArea); //Anexa outputArea ao Conteiner
      int result;
      String output = "";
      for(int counter = 1; counter<=10; counter++){
            result = square(counter);
            output = output + "\n" + counter + " * " + counter + " = " + result;
      }
    outputArea.setText(output);
    }

    public int square(int y){
            return y * y;
    }
}
```

Chamada de métodos

- Métodos da mesma classe:
  - nome_método(lista_parâmetros)
  - se não tiver parâmentros: nome_método()
  - ex: square(counter)
- Métodos de objetos
  - nome_objeto.nome_método();
  - ex: outputArea.setText(“Hello world”);
- Métodos de classes importadas
  - nome_classe.nome_método(lista_parâmetros)
  - ex: JOPtionPane.showMessageDialog(null,“Olá!”);
  - Obs: métodos static
- Um método static de uma classe só pode chamar metodos e variáveis static desta classe

Métodos da Classe Math

- Os métodos dessa classe permitem realizar certos cálculos matemáticos comuns
- Esta presente no pacote java.lang, ou seja, não precisa ser importado
- Seus principais métodos são:

Métodos da Classe Math

- abs(x): retorna o valor absoluto
  - abs(-1.34) = 1.34 (int, long, float, double)
- cos(x): retorna o co-seno de x, em radianos
- exp(x): método exponencial e^x
- ceil(x): retorna o menor inteiro maior que x
  - ceil(9.2) = 10
- floor(x): retorna o maior inteiro menor que x
  - floor(9.2) = 9
- round(x): arredonda para o inteiro mais próximo

Métodos da Classe Math

- log(x): logaritmo de x na base *e*
- max(x,y): retorna o maior entre x e y
- min(x,y): retorna o menor entre x e y
- pow(x,y): retorna x^y
- sin(x): seno de x, em radianos
- sqrt(x): raiz quadrada de x
- tan(x): tangente de x, em radianos

Exercício

- Crie um applet que receba 3 valores (double) e retorne o maior deles.
- Crie um método denominado Max, que será responsável em calcular qual é o maior valor
- Utilize nesse médodo que será criado o método Math.max(x,y).
- O resultado deve ser exibido dentro de uma JTextArea no painel de conteúdo do applet.

```
import javax.swing.*;
import java.awt.Container;

public class Maximum extends JApplet{
    public void init(){
        String n1, n2, n3;
        double x, y, z;
        n1 = JOptionPane.showInputDialog("Entre com o
                                          primeiro número:");
        n2 = JOptionPane.showInputDialog("Entre com o
                                          segundo número:");
        n3 = JOptionPane.showInputDialog("Entre com o
                                          terceiro número:");
```

```
        x = Double.parseDouble(n1);
        y = Double.parseDouble(n2);
        z = Double.parseDouble(n3);
        JTextArea outputArea = new JTextArea();
        outputArea.setText("O maior é " + max(x,y,z));
        Container container = getContentPane();
        container.add(outputArea);
    }
    public double max(double x, double y, double z){
        return Math.max(x,Math.max(y,z));
    }
};
```

Coerção de argumentos

- É a transformação automática de um tipo de dados para outro
- Ex: Math.sqrt(x), espera que x seja double. Se x for inteiro, automaticamente ocorre a conversão de x para double e o método executa normalmente
- Só é aceito de tipos inferiores para tipos superiores

Coerção de argumentos

- double → nenhuma
- float → double
- long → float, double
- int → float, double, long
- char → float, double, long, int
- shor → float, double, long, int
- byte → short, float, double, long, int
- boolean → nenhuma

Conversão de tipos

- Para converter de tipos superiores para inferiores segue-se o exemplo:

```
int x;
x = (int) Max(1.0, 2.0, 3.0)
```

O método retorna um double e esse double é convertido para inteiro

- Exemplo 2:

```
int x = (int) Max(1.0, 2.0, 3.0) / 2
```

Primeiro o método é executado, depois ocorre a conversão e por último a divisão

Números Aleatório

- O método Math.random() retorna um número x (double), de forma que

$0 <= x < 0$

- Para produzir inteiros em um determinado intervalo (I → F) deve-se ***escalonar*** o resultado desse método da seguinte forma:

```
int i = I + (int) (Math.random() * (F – I + 1))
```

Números Aleatórios

```
public class randomico{
    public static void main(String args[]){
        int x=0;
        for (int i=1; i<=10 ; i++){
            x = 1 + (int)(Math.random() * 10);
            System.out.println("\n" + x);
        }
    }
}
```

Exercício

- Crie um método que, quando chamado, gere números inteiros aleatórios de 0 a 6. Crie um aplicativo que receba um inteiro fornecido pelo usuário e compare se esse número é igual a um número gerado pelo método citado acima. Esse aplicativo deve oferecer 10 chances para o usuário acertar. Toda vez que o usuário entrar com um novo valor um novo número aleatório é gerado.

```
import javax.swing.JOptionPane;
public class dado{
    public static int jogar(){
          return 1 + (int)(Math.random()*6);
    }
    public static void main(String args[]){
       String entrada;
       int num, result;
       for (int u=1; u<=10 ; u++){
          entrada = JOptionPane.showInputDialog("Entre com um número (1-6)");
             num = Integer.parseInt(entrada);
             result = jogar();
          if (num == result)
              JOptionPane.showMessageDialog(null,"Você acertou!");
          else
             JOptionPane.showMessageDialog(null,“Errado! O resultado é: " +
    result);
       }
    System.exit(0);
    }
}
```

```
import javax.swing.JOptionPane;
import javax.swing.JApplet;
public class dado extends JApplet{
    public static int jogar(){
          return 1 + (int)(Math.random()*6);
    }
    public void init(){
       String entrada;
       int num, result;
       for (int u=1; u<=10 ; u++){
          entrada = JOptionPane.showInputDialog("Entre com um número (1-6)");
             num = Integer.parseInt(entrada);
             result = jogar();
          if (num == result)
                                  JOptionPane.showMessageDialog(null,"Você acertou!");
          else
            JOptionPane.showMessageDialog(null,"Errado! O resultado é: " + result);
       }
    }
}
```

Exercícios

- Faça um applet que crie um número aleatório entre 0 e 100, e que ofereça 3 chances para o usuário acertar tal número. Toda vez que o usuário entrar com uma tentativa o applet deve retornar se a o número inserido pelo usuário é maior ou menor que o número gerado pelo programa
- Crie um applet onde o usuário insere um número (double) e o applet retorne a parte fracionário deste número
- Crie um applet que gere 5000 números aleatórios de 1 a 5 ao final imprima quantas vezes cada número foi gerado.

Exercício

- Escreva o seguinte método:

public double calculo(double a, double b, int opcao)

Esse método deve retornar o resuldado uma operação entre a e b, que deve ser definida em opcao, da seguinte forma:

  - opcao = 1 → a * b
  - opcao = 2 → a / b
  - opcao = 3 → a^b
  - opcao = 4 → sqrt(a+b)

Crie um applet que utilize esse método e ofereça ao usuário um menu contendo as 4 opções.

Strings

Programação Orientada a Objetos

Java

(Rone Ilídio)

String

- Não é exatamente um vetor de caracteres, é uma classe que possui como atributo um vetor de caracteres.
- Com isso, cada string é um objeto
- Possui vários métodos que auxiliam no tratamento de strings

Strings - Métodos

- Criando um string

```
char[] helloArray = { 'h', 'e', 'l', 'l', 'o', '.'};
String helloString = new String(helloArray);
```

ou

```
String helloString = “hello.”
```

- Comparação

```
String a;
If (a.equals(“Programação”)) { ...}
```

Strings - Métodos

- Tamanho

  String palindrome = "Dot saw I was Tod";

  int len = palindrome.length();

- Concatenando
  - String a = “Hello”
  - String b = “ Word!”
  - a.concat(b); ou a = a+b; ou a +=b;

Strings - Métodos

- Convertendo Strings em números
  - int i = Integer.parseInt(“12”);
  - double d = Double.parseDouble(“3.14”);
- Convertendo número em Strings
  - double x = 10.7;
  - String y = “” + 10.7;

  ou
  - int i; double d;
  - String s3 = Integer.toString(i);
  - String s4 = Double.toString(d);

Strings - Métodos

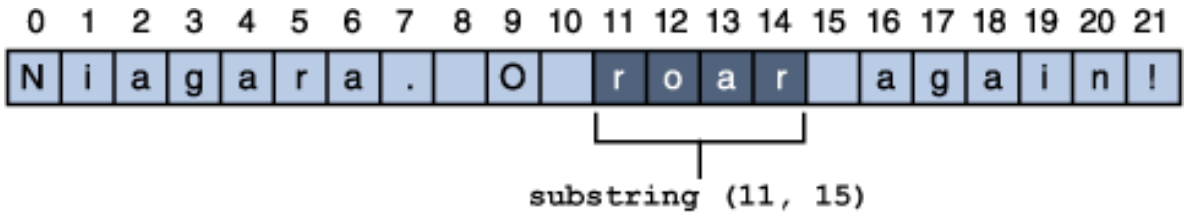
- Pegando um caractere

  String anotherPalindrome = "Niagara. O roar again!";

  char aChar = anotherPalindrome.charAt(9);

Obs: primeiro caractere → charAt(0)

- Pegando uma substring
  - String anotherPalindrome = "Niagara. O roar again!";
  - String roar = anotherPalindrome.substring(11, 15);



String - Métodos

- Maiúsculo e minúsculo
  - toLowerCase() e toUpperCase() : retorna a string toda em maiúsculo ou em minúsculo

- Pegando pedaços da string
  - String question = "rone ilidio da silva";
  - String a[] = question.split(" ");
  - JOptionPane.*showMessageDialog*(**null**, "" + a[0]);

String - Métodos

- Verificando a ocorrência de caracteres ou de substrings
  - String q = "rone ilidio da silva";
  - **int** a = q.indexOf(" ");
  - JOptionPane.*showMessageDialog*(**null**, "" + a); // aparece 4

  - String q = "rone ilidio da silva";
  - **int** a = q.lastIndexOf(" ");
  - JOptionPane.*showMessageDialog*(**null**, "" + a); // aparece 14

  Obs: pode ser passada uma substring

String - Métodos

- Substituindo caracteres ou substrings

```
String q = "rone ilidio da silva";
String a = q.replace(" ","_");
JOptionPane.showMessageDialog(null, "" + a);
```

Formatando casas decimais

```
import java.text.DecimalFormat;
import javax.swing.*;
public class FormataDecimais {
    public static void main(String args[]) {
        double x = 10.0 /3;
        DecimalFormat fmt = new DecimalFormat("0.00");
        String n = fmt.format(x);
        JOptionPane.showMessageDialog(null, "" + n);
    }
}
```