
LTP 1 – Java Básico - Aula 10

Tópicos

- 1 - Solução dos exercícios da aula passada
 - 1.1 - Exercício do CPF
 - 1.2 - Número sorteado
 - 1.3 - Menu de cálculo
 - 1.4 - Projeto Genoma

1 - Solução dos exercícios da aula passada

1.1 - Exercício do CPF

Crie 4 classes dentro do mesmo pacote segundo os critérios abaixo:

- Classe 1: deve ter o método principal, que use o switch para criar ou validar um CPF.
- Classe 2: deve ter dois métodos com o mesmo nome e parâmetros diferentes (sobrecarga de método);
 - Um método deve receber o número de CPF (String), converter para array int e verificar se o mesmo é válido ou não e retornar um valor boolean (true para CPF válido e false para CPF inválido).
 - O outro método não deve receber parâmetro algum, mas deve retornar um número de CPF válido como sendo um array int.
- Classe 3: deve ler um número de CPF, chamar o método que valida o CPF e de acordo com a resposta do tipo boolean, informar se o CPF é válido ou não.
- Classe 4: deve chamar o método que cria um CPF e mostrar o número gerado no formato array int.

```
import java.util.*;
public class CPF
{
    private int numeroCPF[] = new int[11];
    public int[] metodoCPF()
    {
        int cont,soma,resto,dv1,dv2;
        Random x = new Random();
        int i;
        for(i=0;i<11;i++)
            numeroCPF[i]=x.nextInt(9);
        soma=0;
        cont=10;
        for(i=0;i<9;i++)
        {
            soma+=numeroCPF[i]*cont;
            cont--;
        }
        resto=soma%11;
        if(resto<2)
            numeroCPF[9]=0;
        else
            numeroCPF[9]=11-resto;
        soma=0;
```

```
        cont=11;
        for(i=0;i<10;i++)
        {
            soma+=numeroCPF[i]*cont;
            cont--;
        }
        resto=soma%11;
        if(resto<=2)
            numeroCPF[10]=0;
        else
            numeroCPF[10]=11-resto;
        return numeroCPF;
    }
}

public boolean metodoCPF(String valor){
    boolean resultado=false;
    int numeroCPF[] = new int[11];
    int cont,soma,resto,dv1,dv2;
    int i;
    for(i=0;i<11;i++)
        numeroCPF[i]=valor.charAt(i)-48;
    soma=0;
    cont=10;
    for(i=0;i<9;i++)
    {
        soma+=numeroCPF[i]*cont;
        cont--;
    }
    resto=soma%11;
    if(resto<=2)
        dv1=0;
    else
        dv1=11-resto;
    soma=0;
    cont=11;
    for(i=0;i<10;i++)
    {
        soma+=numeroCPF[i]*cont;
        cont--;
    }
    resto=soma%11;
    if(resto<=2)
        dv2=0;
    else
        dv2=11-resto;
    if(dv1==numeroCPF[9]&&dv2==numeroCPF[10])
        return true;
    else
        return false;
}
}
```

```
import java.util.Scanner;
public class Classe {
    public static void main(String args[]){
        int opcao;
        Scanner aux = new Scanner(System.in);
        LerDados x = new LerDados();
        Gerador y = new Gerador();
        System.out.println("Digite 1 para validar um CPF e 2 para criar um CPF");
        opcao = aux.nextInt();
        switch(opcao){
            case 1:x.validaCPF();break;
            case 2:y.criaCPF();break;
            default:System.out.println("Valor inválido");break;
        }
    }
}
```

```
import java.util.Scanner;
public class LerDados {
    public void validaCPF(){
        Scanner aux = new Scanner(System.in);
        CPF x = new CPF();
        String cpfLido;
        boolean valor;
        System.out.println("Entre com seu CPF");
        cpfLido = aux.nextLine();
        if(x.metodoCPF(cpfLido)==true)
            System.out.println("CPF válido");
        else
            System.out.println("CPF inválido");
    }
}
```

```
public class Gerador {
    public void criaCPF(){
        int cpf[] = new int[11];
        int i;
        CPF aux = new CPF();
        cpf=aux.metodoCPF();
        for(i=0;i<11;i++)
            System.out.printf("%d",cpf[i]);
        System.out.println("");
    }
}
```

1.2 - Número sorteado

Crie um jogo para o usuário descobrir um número sorteado de 1 a 100. A cada tentativa dele, forneça uma dica mostrando se o número é maior ou menor. Quando ele descobrir exiba uma mensagem de parabéns e mostre em quantas tentativas ele conseguiu.

```
package classe;
import java.util.*;
public class Classe {
    public static void main(String[] args) {
        Random rand = new Random();
        Scanner ent = new Scanner(System.in);
        int tentativa=0,num=0;
        int sort=rand.nextInt(99);
        while(num!=sort){
            System.out.println("Dê um palpite");
            num=ent.nextInt();
            if(num<sort){
                System.out.println("Tente um número maior");
                tentativa++;
            }
            if(num>sort){
                System.out.println("Tente um número menor");
                tentativa++;
            }
            if(num==sort)
                System.out.printf("Parabéns você acertou depois de %d tentativas",tentativa);
        }
    }
}
```

1.3 - Menu de cálculo

Implementação de aplicação com menu para cálculo de área de figuras trigonométricas. Implemente um programa que atenda aos seguintes requisitos:

- O Programa deverá apresentar um menu com a seguinte interface: Programa de cálculo de áreas de figuras trigonométricas

Opções:

1 – Área de uma circunferência

2 – Área de um triângulo

3 – Sair

Digite sua opção:

- O programa deverá utilizar a biblioteca JOptionPane para as interfaces
- O programa somente poderá ser encerrado quando o usuário selecionar a opção 3

Dicas & Sugestões

- Permita ao usuário selecionar a opção através da digitação de número inteiro
- Armazene o menu em variável String
menu = " Programa de cálculo de áreas de figuras trigonométricas";
menu += "\n\nOpções: \n1 – Área de uma circunferência";

- ```
menu += "\n2 – Área de um triângulo \n3-Sair \n\n Digite sua opção:";
```
- Utilize um laço for infinito para apresentar as opções de menu ao usuário
  - Utilize o método `parseInt` para transformar o valor digitado de string para número inteiro

```
import javax.swing.JOptionPane;
public class Classe {
 public static void main(String[] args) {
 for(;;){
 String valorDigitado =
 JOptionPane.showInputDialog(
 "1 – Área de uma circunferência \n" +
 "2 – Área de um triângulo\n" +
 "3 – Sair");
 int opcao = Integer.parseInt(valorDigitado);
 Calculos aux = new Calculos();
 switch(opcao){
 case 1: aux.calculo1();break;
 case 2: aux.calculo2();break;
 case 3: System.exit(1);break;
 default:
 JOptionPane.showMessageDialog(null,
 "Opcao nao implementada!",
 "Alerta",
 JOptionPane.INFORMATION_MESSAGE);
 break;
 }
 }
 }
}
```

```
import javax.swing.JOptionPane;
public class Calculos {
 public void calculo1(){
 String valor;
 valor = JOptionPane.showInputDialog("Entre com o valor do raio da circunferência");
 double raio=Double.parseDouble(valor);
 double area = Math.PI*Math.pow(raio,2);
 valor = String.format("Área do triângulo = %.2f\n",area);
 JOptionPane.showMessageDialog(null,valor);
 }
 public void calculo2(){
 String valor;
 valor = JOptionPane.showInputDialog("Entre com o valor da altura do triângulo");
 double altura=Double.parseDouble(valor);
 valor = JOptionPane.showInputDialog("Entre com o valor da base do triângulo");
 double base=Double.parseDouble(valor);
 double area = (base*altura)/2;
 valor = String.format("Àrea da circunferência = %.2f\n",area);
 JOptionPane.showMessageDialog(null,valor);
 }
}
```

## 1.4 - Projeto Genoma

Um grande projeto mundial está em curso para mapear todo o material genético do ser humano: o Projeto Genoma Humano. As moléculas de DNA (moléculas que contêm material genético) podem ser representadas por cadeias de caracteres que usam um alfabeto de apenas 4 letras: 'A', 'C', 'T' e 'G'. Um exemplo de uma tal cadeia é:

TCATATGCAAATAGCTGCATACCGA

Nesta tarefa você deverá produzir uma ferramenta muito utilizada no projeto Genoma: um programa que procura ocorrências de uma pequena cadeia de DNA (que vamos chamar de p) dentro de uma outra cadeia de DNA (que vamos chamar de t). Você deverá procurar ocorrências "diretas", que ocorre quando a cadeia p aparece como subcadeia dentro de t. Por exemplo, se

p = CATA

t = TCATATGCAAATAGCTGCATACCGA,

então p ocorre na forma direta na posição 2 e na posição 18 de t.

Tarefa

Sua tarefa é escrever um programa que, dadas duas cadeias p e t, onde o comprimento de p é menor ou igual ao comprimento de t, procura todas as ocorrências diretas. Leia a cadeia p de um array qualquer. Faça uma busca da sequência p na sequência t que deverá estar num segundo array. Por fim, mostre a cadeia p, a cadeia t e a posição das ocorrências diretas.

Exemplo de Entrada

CATA

TCATATGCAAATAGCTGCATACCGA

Exemplo de Saída

CATA

TCATATGCAAATAGCTGCATACCGA

ocorrência direta nas posições: 2 18

---

```
package classe;
public class Classe {
 public static void main(String[] args) {
 String menor="CATA";
 String maior="ACATATGTACCAGGATACATAGTGGA";
 int i=0,cont=1;
 int tam=maior.length(); System.out.println(maior);
 System.out.printf("amostra de tamanho %d\n\n",tam);
 while(i<tam-3)
 {
 if(maior.charAt(i)==menor.charAt(0))
 if(maior.charAt(i+1)==menor.charAt(1))
 if(maior.charAt(i+2)==menor.charAt(2))
 if(maior.charAt(i+3)==menor.charAt(3))
 System.out.printf("sequencia %d localizada na posicao %d\n",cont++,i+1);
 i++;
 }
 }
}
```

---