LTP 1 – Java Básico - Aula 10

Tópicos

1 - Solução dos exercícios da aula passada

1.1 - Conta Bancária

Crie uma classe Conta, que possua um saldo os métodos para pegar saldo, depositar e sacar.

- Crie a classe Conta
- Adicione o atributo saldo
- Crie os métodos getSaldo(), deposita(double) e saca(double)

```
public class Conta {
    private double saldo;
    public void deposita(double valor) {
            this.saldo += valor;
    }
    public void saca(double valor) {
            this.saldo -= valor;
    }
    public double getSaldo() {
            return this.saldo;
    }
  }
```

1.2 - Adicionando um novo método à Conta Bancária

- Adicione um método na classe Conta, que atualiza o saldo dessa conta de acordo com uma taxa percentual fornecida.

```
public class Conta {
    private double saldo;
    public void deposita(double valor) {
            this.saldo += valor;
    }
    public void saca(double valor) {
            this.saldo -= valor;
    }
    public double getSaldo() {
            return this.saldo;
    }
```

```
    public void atualiza(double taxa) {
            this.saldo += this.saldo * taxa;
    }
}
```

1.3 - Subclasses da Conta

Crie duas subclasses da classe Conta: ContaCorrente e ContaPoupanca. Ambas terão o método atualiza reescrito: A ContaCorrente deve atualizar-se com o dobro da taxa e a ContaPoupanca deve atualizar-se com o triplo da taxa.

Além disso, a ContaCorrente deve reescrever o método deposita, a fim de retirar uma taxa bancária de dez centavos de cada depósito.

- Crie as classes ContaCorrente e ContaPoupanca. Ambas são filhas da classe Conta
- Reescreva o método atualiza na classe ContaCorrente, seguindo o enunciado
    - Repare que, para acessar o atributo saldo herdado da classe Conta, **você vai precisar trocar o modificador de visibilidade de saldo para** protected.
- Reescreva o método atualiza na classe ContaPoupanca, seguindo o enunciado
- Na classe ContaCorrente, reescreva o método deposita para descontar a taxa bancária de dez centavos

```
public class ContaPoupanca extends Conta {
    public void atualiza(double taxa) {
            this.saldo += this.saldo * taxa * 3;
    }
}

public class ContaCorrente extends Conta {
    public void atualiza(double taxa) {
            this.saldo += this.saldo * taxa * 2;
    }
    public void deposita(double valor) {
            this.saldo += valor - 0.10;
    }
}
```

1.4 - Crie o método main

Crie uma classe com método main e instancie essas classes, atualize-as e veja o resultado.

- Instancie um objeto para a classe Conta
- Instancie um objeto para a classe ContaCorrente
- Instancie um objeto para a classe ContaPoupanca
- Chame os métodos deposita de cada uma das classes, passando o valor 1000 para cada uma, por exemplo
- Chame os métodos atualiza de cada uma das classes, passando o valor 0.01 para cada uma, por exemplo
- Mostre o saldo a partir de cada uma das classes.

```
public class TestaContas {
	public static void main(String[] args) {
		Conta c = new Conta();
		ContaCorrente cc = new ContaCorrente();
		ContaPoupanca cp = new ContaPoupanca();

		c.deposita(1000);
		cc.deposita(1000);
		cp.deposita(1000);

		c.atualiza(0.01);
		cc.atualiza(0.01);
		cp.atualiza(0.01);

		System.out.println(c.getSaldo());
		System.out.println(cc.getSaldo());
		System.out.println(cp.getSaldo());
	}
}
```

2 - Novos exercícios

2.1 - 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.

2.2 - 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.

2.3 - 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

2.4 - 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 seqüência p na seqüência 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