

Prática 4

1. Considere as classes em Java a seguir.
 - a. O que o programa exibe nos comandos em destaque e porque?
 - b. O que é necessário fazer para que o campo *idadeMinima* da classe *Aluno* seja uma constante?
 - c. Se o campo *idadeMinima* fosse *protected* e as classes *Aluno* e *Principal* estivessem no mesmo pacote, seria necessário utilizar os métodos *alterarIdadeMinima* e *obterIdadeMinima* da classe *Aluno* (considerando que a classe *Aluno* não seria utilizada em outra parte do software)? Por quê? Faça esta alteração no programa e verifique o funcionamento.

```
public class Aluno {  
  
    private String nome;  
    private String matricula;  
    private String situacao;  
    private int idade;  
    private static int idadeMinima = 18;  
  
    public Aluno (String n, String m, int id){  
        nome = n;  
        matricula = m;  
        situacao = "Nao matriculado";  
        if (id >=idadeMinima)  
            idade = id;  
        else  
            idade = 0;  
    }  
  
    public void alterarNome (String n) {  
        nome = n;  
    }  
  
    public String obterNome () {  
        return nome;  
    }  
  
    public void matricular () {  
        situacao = "Matriculado";  
    }  
  
    public void cancelarMatricula () {  
        situacao = "Nao matriculado";  
    }  
  
    public String obterMatricula () {
```

```

    return matricula;
}

public String obterSituacao(){
    return situacao;
}

public void imprimir(){
    System.out.println("** Dados do Aluno **");
    System.out.println("Nome: " + nome);
    System.out.println("Matricula: " + matricula);
    System.out.println("Situação: " + situacao);
    System.out.println("Idade : " + idade);
}

public void alterarIdade(int id){
    if (id >= idadeMinima) idade = id;
}

public int obterIdade(){
    return idade;
}

public void alterarIdadeMinima(int id){
    if (id >0) idadeMinima = id;
}

public int obterIdadeMinima(){
    return idadeMinima;
}
}

```

```

public class Principal {
    public static void main(String[] args){
        Aluno a, b, c;

        a = new Aluno("João", "1", 20);
        a.matricular();
        a.imprimir();

        b = new Aluno("Maria", "2", 22);
        b.imprimir();

        c = new Aluno ("Carlos", "3", 19);
        c.matricular();
        c.imprimir();

        a = c;
        a.imprimir();
        c.imprimir();
        c.alterarIdade(25);
        a.imprimir(); // O que ocorre aqui e porque?

        //O que o programa exibe aqui e porque?
        a.alterarIdadeMinima(15);
    }
}

```

```

        System.out.println("Idade Mínima vista por a e: " +
            a.obterIdadeMinima());
        System.out.println("Idade Mínima vista por b e: " +
            b.obterIdadeMinima());
        System.out.println("Idade Mínima vista por c e: " +
            c.obterIdadeMinima());
    }
}

```

2. O sistema de controle de vendas de uma loja necessita gerenciar dados de produtos, vendedores e clientes, dentre outros.

Implemente uma classe **Produto**. A classe deverá possuir os seguintes atributos:

- código
- descrição
- preço
- quantidade em estoque

A classe deverá possuir um método que permita realizar baixa de estoque. Este método recebe como parâmetro a quantidade a ser retirada do estoque. Caso a quantidade informada seja superior à existente em estoque, a operação não deverá ser realizada. Caso contrário, a quantidade disponível em estoque deverá ser atualizada.

A classe deverá possuir um método para registrar entrada da mercadoria em estoque. Este método recebe como parâmetro a quantidade a ser incluída no estoque e, então, atualiza a quantidade disponível no estoque.

Implemente uma classe **Vendedor**. A classe deverá possuir os seguintes atributos:

- matrícula
- nome
- salário

Implemente uma classe **Cliente**. A classe deverá possuir os seguintes atributos:

- código
- nome
- endereço
- telefone
- tipo (cliente comum ou cliente especial)

Implemente uma aplicação que crie e manipule (altere os dados e os exiba) de três objetos de cada uma das classes. O programa deverá exibir os dados do produto que possui menor quantidade em estoque.