Professor: Edson Ifarraguirre Moreno
Aluno: Ana Carolina de Oliveira Xavier
Turma: 11

Relatório referente ao trabalho final da disciplina
Fundamentos de Programação.

☺ **Introdução:**

O seguinte trabalho tem como objetivo apresentar, com detalhes, as funcionalidades do código fonte referente a um sistema de controle de uma loja. Assim como, expor os conteúdos vistos durante o semestre que contribuíram para a construção do código.

O código é composto por três classes, sendo elas: “Produto”, “Loja” e “Principal”.

☺ **Arquitetura:**

- ## Classe: Produto

A classe “Produto” teve como objetivo a criação de atributos, declarados na classe, e a manipulação dos mesmos através de métodos públicos.

- **Atributos de “Produto”:**
  - ***private String nome**:* Dado privado, do tipo String, para atribuir o nome do produto.
  - ***private double valor:*** Dado privado, do tipo double, para atribuir o valor unitário do produto.
  - ***private int qtEstoque:*** Dado privado, do tipo int, para atribuir a quantidade estocada do produto.
  - ***private int qtVendas:*** Dado privado, do tipo int, para atribuir as quantidades de vendas obtidas do produto.
  - ***private double valorCompras:*** Dado privado, do tipo double, para atribuir o valor total de vendas do produto.

- **Métodos de “Produto”:**

  Os métodos criados nesta classe se comunicam unicamente com a classe “Loja”, portanto, os parâmetros recebidos são através de métodos criados na classe “Loja”, posteriormente explicada neste relatório.

  - ***public Produto():*** Construtor que recebe como parâmetro o nome, valor unitário e quantidade de estoque do produto. No construtor é realizada a atribuição desses valores para cada produto criado.
  - ***public String getNome():*** Método público que retorna o nome do produto.
  - ***public double getValor():*** Método público que retorna o valor unitário do produto.
  - ***public int getQtEstoque():*** Método público que retorna a quantidade de estoque do produto.
  - ***public void CompraRealizada():*** Método público que por meio do parâmetro recebido, nomeado como “compreRe”, manipula os atributos “qtEstoque”, “valorCompras” e “qtVendas” a cada compra realizada no sistema.
  - ***public void QtVendas():*** Método público que manipula o acréscimo da quantidade de vendas.

- ***public int getQtVendas():*** Método público que retorna a quantidade de vendas.
- ***public double getValorCompra():*** Método público que retorna o valor total de compra.
- ***Public int getNovoEstoque():*** Método público que através do parâmetro recebido, nomeado como "novoEstoque", manipula o atributo "qtEstoque" a cada novo registro de estoque realizado no sistema.
- ***public String toString():*** Método público que retorna nome, valor e qtEstoque como frase.

• Classe: Loja

A classe "Loja" teve como objetivo a criação de atributos e objeto, declarados na classe, e a manipulação dos mesmos através de métodos públicos próprios e invocação de métodos do objeto criado.

- **Atributos e objeto de "Loja":**
  - ***private Produto [] listaProdutos = new Produto[100]:*** Criação do objeto listaProdutos, como array, para armazenamento dos produtos cadastrados, limitado até 100 produtos.
  - ***private int nProdutos****:* Dado privado, do tipo int, para atribuir a quantidade de produtos cadastrados.
  - ***private int nProdutosDisponiveis:*** Dado privado, do tipo int, para atribuir a quantidade de produtos disponíveis para compra, ou seja, estoque do produto sendo diferente de 0.
  - ***private int quantidadeProdutos:*** Dado privado, do tipo int, para atribuir a quantidade de produtos vendidos.
  - ***private int quantidadeVendas***: Dado privado, do tipo int, para atribuir a quantidade de vendas realizadas.
  - ***private double compraRealizada:*** Dado privado, do tipo double, para atribuir o valor das compras dos produtos realizadas.
  - ***private double valorVT:*** Dado privado, do tipo double, para atribuir o valor total das compras realizadas.
  - ***private double valorMV:*** Dado privado, do tipo double, para atribuir o valor médio das compras realizadas.

- **Métodos de "Loja":**

Os métodos criados nesta classe recebem, a partir de parâmetros, informações da classe "Principal" onde há contato com o usuário que administrará o sistema de controle da loja. A partir da criação do objeto Produto, esta classe se comunica com a classe "Produto" através da invocação de métodos.

  - ***public boolean cadastraProduto():*** Método público que recebe como parâmetro o nome, valor unitário e a quantidade de estoque do produto cadastrado. A partir do recebimento dessas informações, é criado um novo produto atribuindo as informações para cada um dos atributos do objeto. Conjuntamente com o acréscimo do atributo "nProdutos".

- ***public void identificarProduto():*** Método público que recebe como parâmetro o nome do produto e a quantidade a ser acrescentada no estoque. Inicialmente é realizada a leitura da lista de produtos, logo em seguida é verificado se o valor na posição do vetor lido é diferente de 0, e caso seja válido, é realizada uma nova condição para verificar se esse valor contém o nome informado por paramento. Assim, é passado por parâmetro a quantidade a ser acrescentada ao estoque do produto por meio da invocação do método de novo estoque.
- ***public int visualizarProduto():*** Método público que recebe como parâmetro a opção de visualização desejada. Inicialmente é realizada a verificação se há produtos cadastrados. Havendo produtos cadastrados, é verificado a opção, sendo:
  - opção igual a 1 disponibiliza a lista de todos os produtos cadastrados, assim como o número de produtos;
  - opção igual a 2 disponibiliza a lista de produtos disponíveis, assim como o acréscimo no atributo “nProdutosDisponiveis”, ou também informa que não há produtos disponíveis;
  - opção igual a 3 disponibiliza a lista de produtos indisponíveis, ou informa que não há produtos sem estoque.
- ***public void compraRealizada():*** Método público que recebe como parâmetro o nome do produto e a quantidade de unidades do produto que será comprado. Nesse método é realizada a leitura do vetor duas vezes:
  - Primeira leitura: é invocado o método de compra realizada, sendo passado a quantidade de unidades para a manipulação do objeto produto. Também é invocado o método para o acréscimo da quantidade de vendas. Juntamente com o atributo “quantidadeProdutos” recebendo a quantidade de produtos comprados.
  - Segunda leitura: o atributo “quantidadeVendas” invoca o método “getQtVendas” manipulado; o atributo “compraRealizada” invoca o método “getValorCompra” igualmente manipulado, e por fim, o atributo “valorVT” recebe o valor das compras realizadas.
- ***public int controleLoja():*** Método público que recebe como parâmetro a opção de controle desejada. Verificando a escolha:
  - escolha igual a 1 disponibiliza a quantidade total de produtos vendidos;
  - escolha igual a 2 disponibiliza a quantidade total de vendas;
  - escolha igual a 3 disponibiliza o valor total de vendas;
  - em escolha igual a 4 o atributo “valorMV” recebe o valor total de vendas divido pela quantidade de produtos vendidos, assim realizando e disponibilizando a média das vendas.

• Classe: Principal

A classe “Principal” teve como objetivo a criação de variáveis e objetos. Esta classe é o meio de comunicação do usuário com o sistema. A Principal se comunicará com a Loja, e a Loja se comunicará com o Produto, não sendo possível se comunicar com o Produto diretamente pela Principal.

- **Variáveis, objetos e métodos de "Principal":**
    - **Método "Main()":**

      Como o nome induz, este é o método principal da classe Principal, a partir deste é realizado um *loop* (do while) com as opções disponíveis para o usuário (switch). Em cada *case* é informado por parâmetro as informações necessárias para a invocação dos métodos do objeto Loja.
        - ***Scanner entrada():*** Criação do objeto Scanner, para entrada de informação do usuário no sistema.
        - ***Loja trabalhoFinal():*** Criação do objeto Loja, para comunicação com a classe Loja e invocação de seus métodos.
        - ***int opção:*** Variável, do tipo int, para manipulação do menu (switch) de escolha do usuário.
        - ***String nomeInf:*** Variável, do tipo String, para a recepção de nome para a realização do cadastro do produto.
        - ***double valorInf:*** Variável, do tipo double, para a recepção de valor unitário para a realização do cadastro do produto.
        - ***int qtEstoqueInf:*** Variável, do tipo int, para a recepção de quantidade de estoque para a realização do cadastro do produto.
        - ***int novoEstoque:*** Variável, do tipo int, para a recepção de quantidade de acréscimo no estoque e realização de registro de estoque.
        - ***int compraRe:*** Variável, do tipo int, para a recepção de quantidade de unidades de produto que o usuário deseja comprar.

    - **Método "menuPrincipal()":**

      Neste método é realizado o *loop* (do while) com as opções principais disponíveis para o usuário. Tem como retorno a opção desejada, resultado do *case* referente no switch no método Main.
        - ***Scanner entrada():*** Criação do objeto Scanner, para entrada de informação do usuário no método.
        - ***int opção:*** Variável, do tipo int, para manipulação do menu (do while) de escolha do usuário.
    - **Método "menuListarProdutos()":**

      Caso a opção do usuário seja igual a 3, será chamado este método. Este realiza o *loop* (do while) com as opções de listagem de produtos disponíveis para o usuário. Tem como retorno a opção desejada, resultado do *case* referente no switch do case 3.
        - ***Scanner entrada():*** Criação do objeto Scanner, para entrada de informação do usuário no método.
        - ***int opção:*** Variável, do tipo int, para manipulação do menu (do while) de escolha do usuário.
    - **Método "menuRegistrarCompra()":**

      Caso a opção do usuário seja igual a 4, será chamado este método. Este realiza o *loop* (do while) com as opções de registrar compra disponíveis. Tem como retorno a opção desejada, resultado do *case* referente no switch do case 4.

- ***Scanner entrada():*** Criação do objeto Scanner, para entrada de informação do usuário no método.
- ***int opção:*** Variável, do tipo int, para manipulação do menu (do while) de escolha do usuário.

- **Método "menuCompra()":**

  Este método é chamado quando o usuário escolhe a opção 4. O menuCompra tem como objetivo a disponibilidade do usuário continuar comprando sem precisar sair da opção de registrar compra. Tem como retorno a opção desejada, esta que modifica o valor da opção do usuário do *case* 4 no switch.

  - ***Scanner entrada():*** Criação do objeto Scanner, para entrada de informação do usuário no método.
  - ***int opção:*** Variável, do tipo int, para manipulação da escolha do usuário.
- **Método "menuControleLoja()":**

  Caso a opção do usuário seja igual a 5, será chamado este método. Este realiza o *loop* (do while) com as opções de controle da loja disponíveis. Tem como retorno a opção desejada, resultado do *case* referente no switch do case 5.

  - ***Scanner entrada():*** Criação do objeto Scanner, para entrada de informação do usuário no método.
  - ***int opção:*** Variável, do tipo int, para manipulação do menu (do while) de escolha do usuário.

☺ **Validação**:

Cadastrando produtos:

```
Menu de opções
1 - Cadastrar produtos
2 - Registrar estoque
3 - Visualização de produtos
4 - Registro de compra
5 - Controle da loja
0 - Sair do programa
Opção desejada: 1

  >> Cadastrar produto
     Informe o nome do produto: Chocolate
     Informe o valor unitário do produto: 4.79
     Informe a quantidade estocada: 50

Menu de opções
1 - Cadastrar produtos
2 - Registrar estoque
3 - Visualização de produtos
4 - Registro de compra
5 - Controle da loja
0 - Sair do programa
Opção desejada: 1

  >> Cadastrar produto
     Informe o nome do produto: Bolacha
     Informe o valor unitário do produto: 2.45
     Informe a quantidade estocada: 35
```

Registrando novo estoque e listando todos os produtos:

```
Menu de opções
1 - Cadastrar produtos
2 - Registrar estoque
3 - Visualização de produtos
4 - Registro de compra
5 - Controle da loja
0 - Sair do programa
Opção desejada: 2

  >> Registrar estoque
     Informe o nome do produto: Chocolate
     Informe a quantidade de estoque entrando: 5
Dados de estoque registrado

Menu de opções
1 - Cadastrar produtos
2 - Registrar estoque
3 - Visualização de produtos
4 - Registro de compra
5 - Controle da loja
0 - Sair do programa
Opção desejada: 3

  >> Lista de produtos
1 - Listar todos os produtos
2 - Listar produtos disponíveis
3 - Listar produtos com estoque zero
0 - Voltar para o menu principal
Opção desejada: 1
Há 2 produtos cadastrados.
     Produto_00 (Chocolate): R$ 4,79 Estoque: 55 unidades.
     Produto_01 (Bolacha): R$ 2,45 Estoque: 35 unidades.
```

Realizando uma compra e mostrando produtos disponíveis e indisponíveis:

```
Menu de opções
1 - Cadastrar produtos
2 - Registrar estoque
3 - Visualização de produtos
4 - Registro de compra
5 - Controle da loja
0 - Sair do programa
Opção desejada: 4

  >> Registrar compra
1 - Selecionar produtos
2 - Finalizar compra
Opção desejada: 1

Produtos disponíveis:
     Produto_00 (Chocolate): R$ 4,79 Estoque: 55 unidades.
     Produto_01 (Bolacha): R$ 2,45 Estoque: 35 unidades.
     Informe o nome do produto: Bolacha
     Informe a quantidade de unidades: 35

1 - Desejo realizar nova compra
2 - Não desejo realizar nova compra
2
Compra finalizada!
Menu de opções
1 - Cadastrar produtos
2 - Registrar estoque
3 - Visualização de produtos
4 - Registro de compra
5 - Controle da loja
0 - Sair do programa
Opção desejada: 3

  >> Lista de produtos
1 - Listar todos os produtos
2 - Listar produtos disponíveis
3 - Listar produtos com estoque zero
0 - Voltar para o menu principal
Opção desejada: 2
     Produto_00 (Chocolate): R$ 4,79 Estoque: 55 unidades.


  >> Lista de produtos
1 - Listar todos os produtos
2 - Listar produtos disponíveis
3 - Listar produtos com estoque zero
0 - Voltar para o menu principal
Opção desejada: 3
Não há produtos sem estoque.
     Produto_01 (Bolacha): R$ 2,45 Estoque: 00 unidades.
```

Acessando controle do sistema da loja:

```
Menu de opções
1 - Cadastrar produtos
2 - Registrar estoque
3 - Visualização de produtos
4 - Registro de compra
5 - Controle da loja
0 - Sair do programa
Opção desejada: 5

  >> Controle da loja
1 - Quantidade de produtos vendidos
2 - Quantidade de vendas
3 - Valor total das vendas
4 - Valor médio das vendas
0 - Voltar para o menu principal
Opção desejada: 1
Quantidade total de produtos vendidos: 35
  >> Controle da loja
1 - Quantidade de produtos vendidos
2 - Quantidade de vendas
3 - Valor total das vendas
4 - Valor médio das vendas
0 - Voltar para o menu principal
Opção desejada: 2
Quantidade total de vendas:  1
  >> Controle da loja
1 - Quantidade de produtos vendidos
2 - Quantidade de vendas
3 - Valor total das vendas
4 - Valor médio das vendas
0 - Voltar para o menu principal
Opção desejada: 3
Valor total das vendas: 85,75
  >> Controle da loja
1 - Quantidade de produtos vendidos
2 - Quantidade de vendas
3 - Valor total das vendas
4 - Valor médio das vendas
0 - Voltar para o menu principal
Opção desejada: 4
Valor médio das vendas: 2,45
```

☺ **Conclusão:** Para a elaboração deste código foram utilizados os aprendizados de declaração de variável, utilização de laços de repetição, condição, criação de métodos, chamada de métodos, criação e manipulação de array e orientação a objetos, assuntos abordados e praticados ao longo do semestre.

☺

No trecho:

```
Scanner entrada = new Scanner(System.in);
Loja trabalhoFinal = new Loja();
int opcao;

String nomeInf;
double valorInf;
int novoEstoque, qtEstoqueInf, compraRe;
```

É possível observar: a criação de objetos (Scanner e Loja). Assim como é possível observar a declaração de variáveis tipo String, double e int.

No trecho:

```
do{
    escolhaLista = menuListarProdutos();
    switch(escolhaLista){
        case 1:
            trabalhoFinal.visualizarProdutos(escolhaLista);
            break;
        case 2:
            trabalhoFinal.visualizarProdutos(escolhaLista);
            break;
        case 3:
            trabalhoFinal.visualizarProdutos(escolhaLista);
            break;
        case 0:
            break;
    }
}while(escolhaLista != 0);
```

É possível observar: a utilização do laço de repetição *do while*, a chamada do método “menuListarProdutos()”, a utilização do *switch* e a invocação do método “visualizarProdutos()” do objeto trabalhoFinal.

No trecho:

```
public Produto(String nomeInf, double valorInf, int qtEstoqueInf){
    nome = nomeInf;
    if(valorInf > 0){
        valor = valorInf;
    }else{
        System.out.println("Valor inválido. Valor padronizado em $0.");
    }
    if(qtEstoqueInf > 0){
        qtEstoque = qtEstoqueInf;
    }else{
        System.out.println("Valor inválido. Estoque padronizado em 0.");
    }
}
```

É possível observar: a criação de um construtor para a manipulação de atributos (valor, qtEstoque e nome), também sendo possível observar a utilização da condição *if/if else*.

No trecho:

```
public class Loja
{
    private int nProdutos, nProdutosDisponiveis;
    private double quantidadeProdutos, quantidadeVendas;
    private double compraRealizada, valorVT, valorMV;
    private Produto [] listaProdutos = new Produto[100];

    public boolean cadastraProduto(String nome, double valor, int qtEstoque){
        if(nProdutos < 100){
            listaProdutos[nProdutos] = new Produto(nome, valor, qtEstoque);
            nProdutos++;
            return true;
        }else{
            return false;
        }
    }
    public void identificarProduto(String nomeInf, int nvEstoque){
        Produto resultado = null;
        for(int pos = 0; pos < nProdutos; pos++){
            if(listaProdutos[pos] != null){
                if(listaProdutos[pos].getNome().toUpperCase().contains(nomeInf.toUpperCase())){
                    listaProdutos[pos].getNovoEstoque(nvEstoque);
                }
            }
        }
        if(nvEstoque != 0){
            System.out.println("Dados de estoque registrado");
            for(int pos = 0; pos < nProdutos; pos++){
                if(listaProdutos[pos].getNome() == nomeInf){
                    System.out.println(listaProdutos[pos].toString());
                }
            }
            }else{
                System.out.println("O estoque do produto " + nomeInf + " não foi registrado.");
            }
    }
```

Neste trecho conseguimos observar a declaração de atributos de uma classe, a criação de array, criação de métodos públicos, criação de objeto, e a manipulação tanto dos objetos quanto do array.

Portanto, pode-se constatar que o semestre foi de grande aprendizado e foi possível colocar em prática toda a teoria ensinada. Neste trabalho final é possível encontrar todos os tópicos apresentados em aula, sendo possível desenvolver um código funcional e organizado.