

1ª VERIFICAÇÃO DE APRENDIZAGEM (PRÁTICA)

Instruções gerais:

Utilize um diretório (do pendrive) para salvar as implementações. Salve as implementações a cada modificação, caso aconteça alguma falha de energia o trabalho será preservado. Lembre-se que uma vez removido o arquivo do Eclipse, seu conteúdo será perdido.

A prova é prática e deverão ser devolvidas tanto a prova impressa quanto os códigos-fonte implementados em Java. A Nota máxima desta prova é de 10,0 pontos.

Criação e Organização de Projeto (0,25 pontos)

- 1) No Eclipse limpe todos os projetos existentes.

- Crie um novo projeto chamado

br.1va.mpoo.edu.NomeSobrenome

Este deverá ter uma pasta de pacotes chamada **supermercado** contendo todos os arquivos necessários para as respectivas questões.

- Ao finalizar a prova **compacte o projeto** contendo toda a codificação do projeto (arquivos textos e *bytecodes*) e envie-o no AVA:

[1ª Verificação de Aprendizagem] em Semana 11

- 2) O proprietário do supermercado MPOO Supermarket precisa de um sistema para seu empreendimento, como, por exemplo, gerenciar produtos, compras e funcionários. O sistema é descrito nas questões abaixo, modelado no APÊNDICE A e deverá ser implementado em Java.

Organização em pacotes (0,25 pontos)

- a) Organize seu projeto conforme os pacotes definidos.

Atributos e métodos static (0,25 pontos)

- b) Faça o devido uso de static em atributos e métodos.

Classe, atributo e método construtor (0,75 pontos)

- c) Faça a devida criação de classes, atributos e métodos construtores.

- d) É descrição do sistema (Apêndice A):

Encapsulamento e métodos de acesso (0,5 ponto)

- Observe os encapsulamento dos atributos.

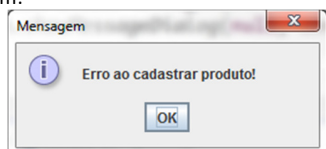
Agregação e ArrayList, definição de métodos, implementações e utilização (1,0 ponto)

- Os produtos do sistema estão armazenados na BaseDados. Faça o devido uso de ArrayList e dos métodos definidos para manipular a base:

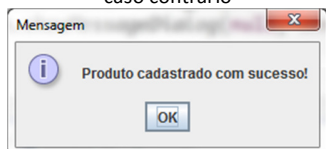
BaseDados
- produtos : ArrayList<Produto>
+ createBase() : void
+ inicializarBase() : void
+ buscarProduto(codBarras : String) : Produto
+ isProduto(codBarras : String) : boolean
+ addProduto(produto : Produto) : boolean
+ removerProduto(produto : Produto) : boolean
+ exibirProdutos() : ArrayList<String>

Nesse sistema cada produto é diferenciado pelo codBarras (único entre os produtos).

- Se um produto já cadastrado tentar ser inserido novamente ou se estiver vencido deve-se exibir uma caixa de diálogo com a mensagem:



caso contrário



- Essa verificação deverá ser realizada no método **public static boolean addProduto(Produto produto){}**, responsável por adicionar um produto no ArrayList. Esse método deverá fazer uso de outro(s) método(s), evitando a duplicidade de código.
- A validade do produto é verificada pelo serviço **isVencido** de Verificador. Portanto, não há produtos vencidos na BaseDados do sistema do supermercado.

Classes e troca de mensagens (0,25 pontos)

Fique Atento!

Você sabe como definir uma data em seu sistema? **Calendar** é classe de **java.util** que permite armazenar datas.

Mas o porquê de não usar **Date** de **java.util**? A resposta é simples: *"Date is deprecated"*.

Então observe o exemplo de codificação de verificação do prazo de validade de um produto:

```
public static boolean isVencido(Produto produto){
    Calendar dataAtual = Calendar.getInstance();
    if(produto.getValidade().before(dataAtual))
        return true;
    return false;
}
```

Mas e como fazemos para criar uma data em um sistema? Por exemplo, para criar a data 11/12/2023 fazemos:

```
Calendar data = Calendar.getInstance();
data.set(2023, Calendar.DECEMBER, 11);
```

- A exibição dos produtos armazenados deverá ser por **public static ArrayList<String> exibirProduto(){}**. Mas apenas os nomes poderão ser informados.
- Implemente os demais métodos presentes na BaseDados.

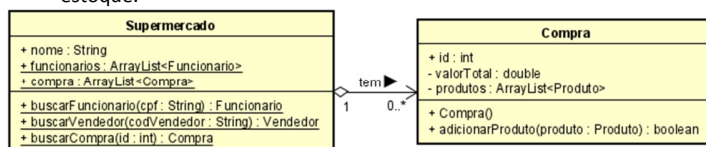
Classes e troca de mensagens (0,25 pontos)

- Todas as caixas de diálogos do sistema devem utilizar o método **exibirMensagem()** de **Mensagem**:

Mensagem
+ PRODUTO_SUCESSO : String = "Produto cadastrado com sucesso" (readOnly)
+ PRODUTO_ERRO : String = "Erro ao cadastrar produto" (readOnly)
+ exibirMensagem(mensagem : String) : void

Agregação e ArrayList, definição de métodos, implementações e utilização (1,0 ponto)

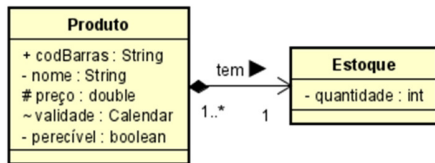
- Todos os produtos armazenados na BaseDados poderão ser comprados em uma Compra do supermercado "MPOO Supermarket", mas se houver quantidade do produto disponível em estoque.



- Todo produto adicionado em uma compra deverá ser removido do estoque (Regra de negócio definido pelo Caixa).
- Uma compra possui um identificador único que representa o número da compra efetuada, sendo este valor nunca repetido e crescente (*auto-increment*). Logo, não é permitido haver duas compras com a mesma identificação.

Composição (0,75 pontos)

- A relação de Produto e Estoque é de composição.



Classe e método abstract (0,5 pontos)

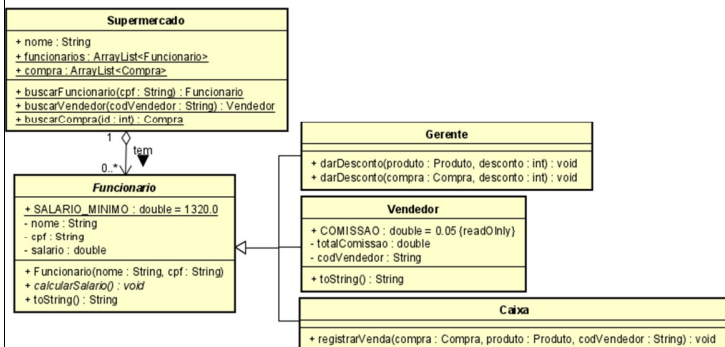
Herança (0,75 pontos)

Polimorfismo de Objetos (0,5 pontos)

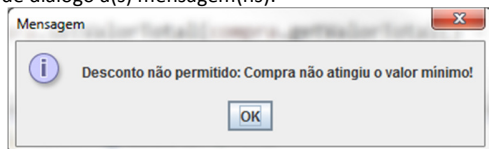
Polimorfismo de Método (0,25 pontos)

Sobrescrita de métodos (0,25 pontos)

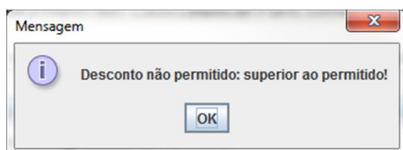
- O supermercado possui funcionários especializados do tipo Gerente, Vendedor e Caixa. Todos os funcionários são registrados em Supermercado. A manipulação dos funcionários deverá utilizar os métodos de ArrayList.



- Um gerente possui um salário de 5 salários mínimos e poderá dar descontos no valor de uma Compra ou no preço de um Produto da base. Descontos poderão ser informados apenas ao final da compra. O desconto máximo é 10% sobre o valor total de uma compra, desde que essa compra possua um valor superior a R\$ 100,00. Caso tente-se burlar o desconto, o sistema exibirá em uma caixa de diálogo a(s) mensagem(ns):



OU



- Um vendedor poderá realizar vendas de produtos. Tem salário inicial de um salário mínimo, mas que poderá ser aumentado a partir das comissões sobre produtos vendidos, para isso seu codVendedor deverá ser informado no ato do registro de uma venda.
- Um caixa recebe um salário mínimo. E tem a função de registrar os produtos de uma compra vendidos por um vendedor.
- É possível ver os dados de um funcionário por toString().

Vinculação dinâmica de método (0,5 pontos)

- O salário de um funcionário é calculado pelo método calcularSalario(). Faça o devido uso de vinculação dinâmica de método.

Documentação (0,5 pontos)

- e) Adicione o java comments para que o desenvolvedor saiba do que se trata os métodos darDesconto de Gerente.

darDesconto

```
public void darDesconto(Produto produto, int desconto)
```

Este método dar o desconto em porcentagem a um produto.

Parameters:

desconto - Desconto em porcentagem. Ex: Se preço de produto é de R\$10,00 e desconto é de 10%, então o novo preço de produto é de R\$ 9,00.

produto - Produto que receberá o desconto.

e

darDesconto

```
public void darDesconto(Compra compra, int desconto)
```

Parameters:

desconto - Desconto em porcentagem. Máximo de 10% para uma compra mínima de R\$ 100,00. Ex: Se valor de compra é de R\$100,00 e desconto é de 10%, então o novo valor da compra é de R\$ 90,00.

compra - Compra que receberá o desconto.

Instâncias e chamadas de métodos (2,0 pontos)

- f) Simule em App, a criação e utilização das instâncias:

- Produtos já cadastrados na base:

- nome: feijão da Serra, codBarras: PROD001, preço: R\$ 10,00, validade de 31/12/2024; não perecível
- nome: arroz da Serra, codBarras: PROD002, preço: R\$ 5,00, validade de 01/06/2024; não perecível
- nome: biscoito formoso, codBarras: PROD003, preço: R\$ 2,00, validade de 10/12/2023; não perecível
- nome: laranja mimosas, codBarras: PROD004, preço: R\$ 0,50, validade de 21/12/2023; perecível.

- Funcionários:

- A gerente Maria Silva (cpf 111.111.111-11) de salário R\$ 6.660,00 (cinco salários mínimo).
- O vendedor João Santos (cpf 222.222.222-22) de salário R\$ 1.320 mais comissões. Seu código: "vend001".
- O vendedor Godofredo Sá (cpf 333.333.333-33) de salário R\$ 1.320 mais comissões. Seu código: "vend002".
- O caixa José Lima (cpf 444.444.444-44) que recebe um salário mínimo.
- Exiba os dados dos funcionários do supermercado no console.

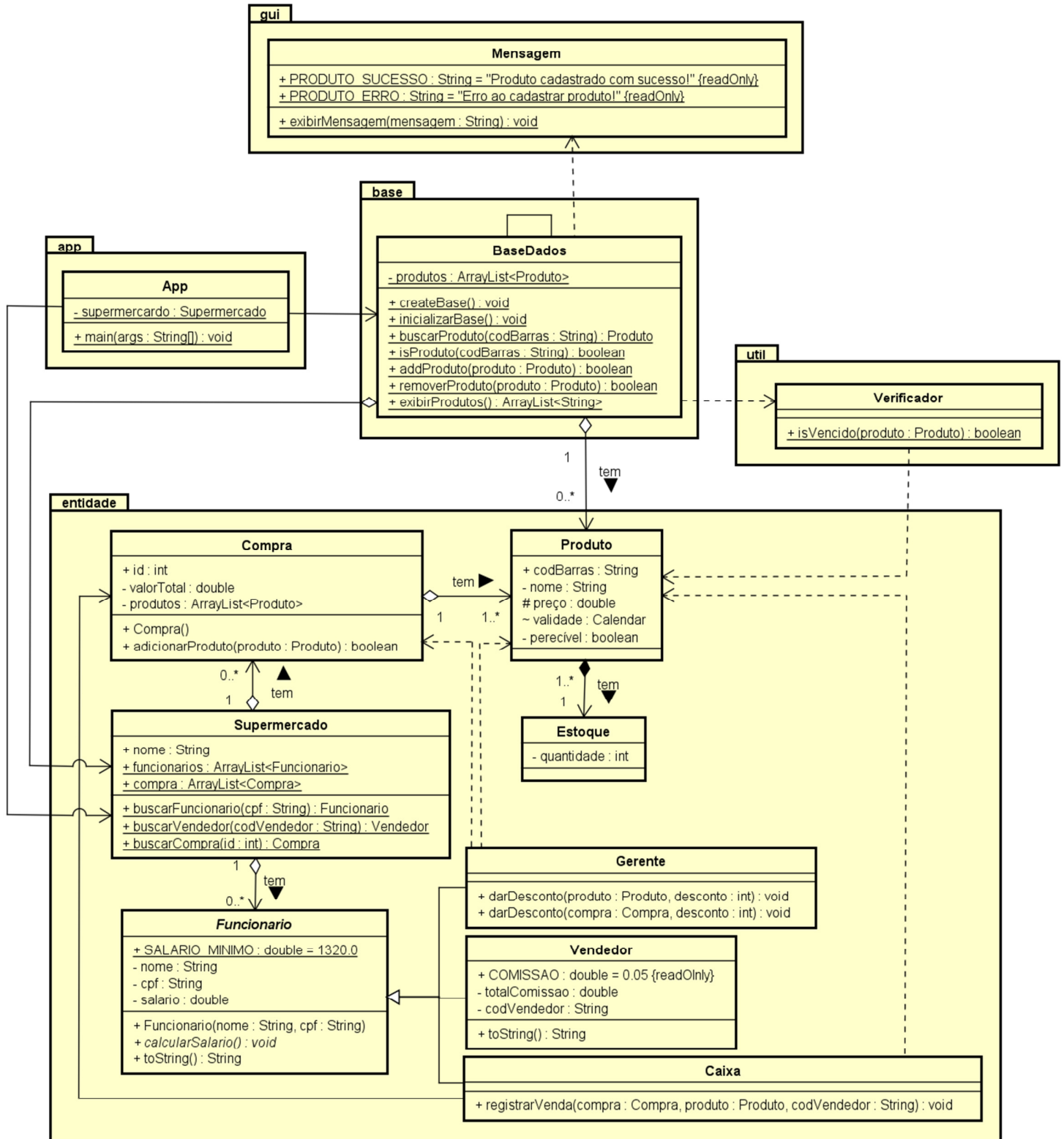
- As compras:

- A primeira com apenas um feijão realizada por João Santos.
- A segunda com 10 unidades de feijão por João Santos.
- Na terceira compra, o caixa precisa registrar apenas um arroz, mas recebeu a indicação do cliente que foi o vendedor "vend003".
- A quarta compra feita por Godofredo Sá de: 10 unidades de feijão e 10 unidades de arroz. Entretanto, o cliente solicitou um desconto de 10%.
- Na quinta compra o cliente gostaria de 10 laranjas mimosas. Mas percebeu que as laranjas estavam começando a estragar procurou o gerente para ele conceder um desconto de 50%. O gerente sabendo dessa situação passou a conceder o desconto para todo cliente. Então o caixa registrou a compra do cliente, mas sem atribuir uma comissão a um vendedor.
- Após essas compras, exiba no console:
 - Todas as compras efetuadas, mostrando no console o id e o respectivo valor.
 - As quantidades disponíveis dos produtos.
 - Os dados dos vendedores.

- 3) Durante a realização da 1ª Verificação de Aprendizagem:
- Qual assunto você teve mais dificuldade (se possível descrever com situação da prova)?
 - Qual assunto você gostaria que tivesse maior pontuação?

```
public class MP00 {
    void saudacao(){OptionPane.showMessageDialog(null, "Boa Prova!");}
    public static void main(String[] args) {
        new MP00().saudacao();
    }
}
```

APÊNDICE A



1ª VERIFICAÇÃO DE APRENDIZAGEM (PRÁTICA)

Instruções gerais:

Utilize um diretório (do pendrive) para salvar as implementações. Salve as implementações a cada modificação, caso aconteça alguma falha de energia o trabalho será preservado. Lembre-se que uma vez removido o arquivo do Eclipse, seu conteúdo será perdido.

A prova é prática e deverão ser devolvidas tanto a prova impressa quanto os códigos-fonte implementados em Java. A Nota máxima desta prova é de 10,0 pontos.

Criação e Organização de Projeto (0,25 pontos)

- 1) No Eclipse limpe todos os projetos existentes.

- Crie um novo projeto chamado

br.1va.mpoo.edu.NomeSobrenome

Este deverá ter uma pasta de pacotes chamada **supermercado** contendo todos os arquivos necessários para as respectivas questões.

- Ao finalizar a prova **compacte o projeto** contendo toda a codificação do projeto (arquivos textos e *bytecodes*) e envie-o no AVA:

[1ª Verificação de Aprendizagem] em Semana 11

- 2) O proprietário do supermercado MPOO Supermarket precisa de um sistema para seu empreendimento, como, por exemplo, gerenciar produtos, compras e funcionários. O sistema é descrito nas questões abaixo, modelado no APÊNDICE A e deverá ser implementado em Java.

Organização em pacotes (0,25 pontos)

- a) Organize seu projeto conforme os pacotes definidos.

Atributos e métodos static (0,25 pontos)

- b) Faça o devido uso de static em atributos e métodos.

Classe, atributo e método construtor (0,75 pontos)

- c) Faça a devida criação de classes, atributos e métodos construtores.

- d) É descrição do sistema (Apêndice A):

Encapsulamento e métodos de acesso (0,5 ponto)

- Observe os encapsulamento dos atributos.

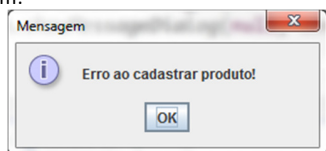
Agregação e ArrayList, definição de métodos, implementações e utilização (1,0 ponto)

- Os produtos do sistema estão armazenados na BaseDados. Faça o devido uso de ArrayList e dos métodos definidos para manipular a base:

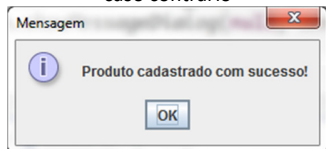
BaseDados
- produtos : ArrayList<Produto>
+ createBase() : void
+ inicializarBase() : void
+ buscarProduto(codBarras : String) : Produto
+ isProduto(codBarras : String) : boolean
+ addProduto(produto : Produto) : boolean
+ removerProduto(produto : Produto) : boolean
+ exibirProdutos() : ArrayList<String>

Nesse sistema cada produto é diferenciado pelo codBarras (único entre os produtos).

- Se um produto já cadastrado tentar ser inserido novamente ou se estiver vencido deve-se exibir uma caixa de diálogo com a mensagem:



caso contrário



- Essa verificação deverá ser realizada no método **public static boolean addProduto(Produto produto){}**, responsável por adicionar um produto no ArrayList. Esse método deverá fazer uso de outro(s) método(s), evitando a duplicidade de código.
- A validade do produto é verificada pelo serviço **isVencido** de Verificador. Portanto, não há produtos vencidos na BaseDados do sistema do supermercado.

Classes e troca de mensagens (0,25 pontos)

Fique Atento!

Você sabe como definir uma data em seu sistema? **Calendar** é classe de **java.util** que permite armazenar datas.

Mas o porquê de não usar **Date** de **java.util**? A resposta é simples: *"Date is deprecated!"*.

Então observe o exemplo de codificação de verificação do prazo de validade de um produto:

```
public static boolean isVencido(Produto produto){
    Calendar dataAtual = Calendar.getInstance();
    if(produto.getValidade().before(dataAtual))
        return true;
    return false;
}
```

Mas e como fazemos para criar uma data em um sistema? Por exemplo, para criar a data 11/12/2023 fazemos:

```
Calendar data = Calendar.getInstance();
data.set(2023, Calendar.DECEMBER, 11);
```

- A exibição dos produtos armazenados deverá ser por **public static ArrayList<String> exibirProduto(){}.** Mas apenas os nomes poderão ser informados.
- Implemente os demais métodos presentes na BaseDados.

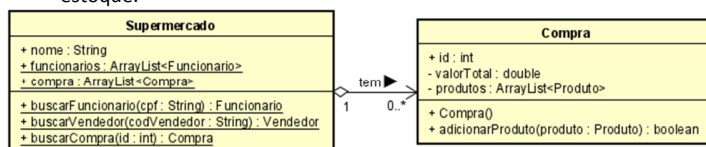
Classes e troca de mensagens (0,25 pontos)

- Todas as caixas de diálogos do sistema devem utilizar o método **exibirMensagem()** de **Mensagem**:

Mensagem
+ PRODUTO_SUCESSO : String = "Produto cadastrado com sucesso" (readOnly)
+ PRODUTO_ERRO : String = "Erro ao cadastrar produto" (readOnly)
+ exibirMensagem(mensagem : String) : void

Agregação e ArrayList, definição de métodos, implementações e utilização (1,0 ponto)

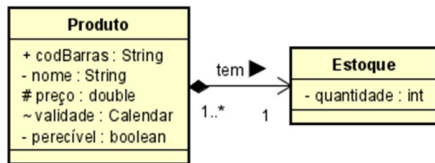
- Todos os produtos armazenados na BaseDados poderão ser comprados em uma Compra do supermercado "MPOO Supermarket", mas se houver quantidade do produto disponível em estoque.



- Todo produto adicionado em uma compra deverá ser removido do estoque (Regra de negócio definido pelo Caixa).
- Uma compra possui um identificador único que representa o número da compra efetuada, sendo este valor nunca repetido e crescente (*auto-increment*). Logo, não é permitido haver duas compras com a mesma identificação.

Composição (0,75 pontos)

- A relação de Produto e Estoque é de composição.



Classe e método abstract (0,5 pontos)

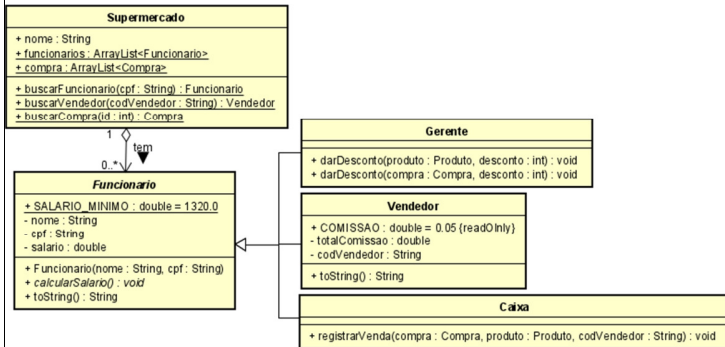
Herança (0,75 pontos)

Polimorfismo de Objetos (0,5 pontos)

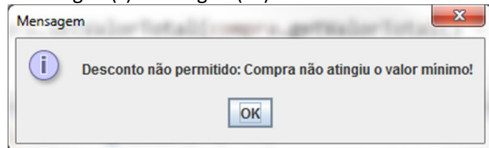
Polimorfismo de Método (0,25 pontos)

Sobrescrita de métodos (0,25 pontos)

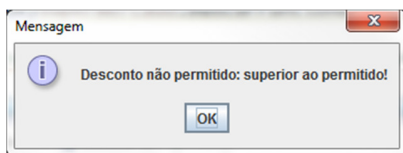
- O supermercado possui funcionários especializados do tipo Gerente, Vendedor e Caixa. Todos os funcionários são registrados em Supermercado. A manipulação dos funcionários deverá utilizar os métodos de ArrayList.



- Um gerente possui um salário de 5 salários mínimos e poderá dar descontos no valor de uma Compra ou no preço de um Produto da base. Descontos poderão ser informados apenas ao final da compra. O desconto máximo é 10% sobre o valor total de uma compra, desde que essa compra possua um valor superior a R\$ 100,00. Caso tente-se burlar o desconto, o sistema exibirá em uma caixa de diálogo a(s) mensagem(ns):



OU



- Um vendedor poderá realizar vendas de produtos. Tem salário inicial de um salário mínimo, mas que poderá ser aumentado a partir das comissões sobre produtos vendidos, para isso seu codVendedor deverá ser informado no ato do registro de uma venda.
- Um caixa recebe um salário mínimo. E tem a função de registrar os produtos de uma compra vendidos por um vendedor.
- É possível ver os dados de um funcionário por toString().

Vinculação dinâmica de método (0,5 pontos)

- O salário de um funcionário é calculado pelo método calcularSalario(). Faça o devido uso de vinculação dinâmica de método.

Documentação (0,5 pontos)

- e) Adicione o java comments para que o desenvolvedor saiba do que se trata os métodos darDesconto de Gerente.

darDesconto

```
public void darDesconto(Produto produto, int desconto)
```

Este método dar o desconto em porcentagem a um produto.

Parameters:

desconto - Desconto em porcentagem. Ex: Se preço de produto é de R\$10,00 e desconto é de 10%, então o novo preço de produto é de R\$ 9,00.

produto - Produto que receberá o desconto.

e

darDesconto

```
public void darDesconto(Compra compra, int desconto)
```

Parameters:

desconto - Desconto em porcentagem. Máximo de 10% para uma compra mínima de R\$ 100,00. Ex: Se valor de compra é de R\$100,00 e desconto é de 10%, então o novo valor da compra é de R\$ 90,00.

compra - Compra que receberá o desconto.

Instâncias e chamadas de métodos (2,0 pontos)

- f) Simule em App, a criação e utilização das instâncias:

- Produtos já cadastrados na base:

- nome: feijão da Serra, codBarras: PROD001, preço: R\$ 10,00, validade de 31/12/2024; não perecível
- nome: arroz da Serra, codBarras: PROD002, preço: R\$ 5,00, validade de 01/06/2024; não perecível
- nome: biscoito formoso, codBarras: PROD003, preço: R\$ 2,00, validade de 10/12/2023; não perecível
- nome: laranja mimosas, codBarras: PROD004, preço: R\$ 0,50, validade de 21/12/2023; perecível.

- Funcionários:

- A gerente Maria Silva (cpf 111.111.111-11) de salário R\$ 6.660,00 (cinco salários mínimo).
- O vendedor João Santos (cpf 222.222.222-22) de salário R\$ 1.320 mais comissões. Seu código: "vend001".
- O vendedor Godofredo Sá (cpf 333.333.333-33) de salário R\$ 1.320 mais comissões. Seu código: "vend002".
- O caixa José Lima (cpf 444.444.444-44) que recebe um salário mínimo.
- Exiba os dados dos funcionários do supermercado no console.

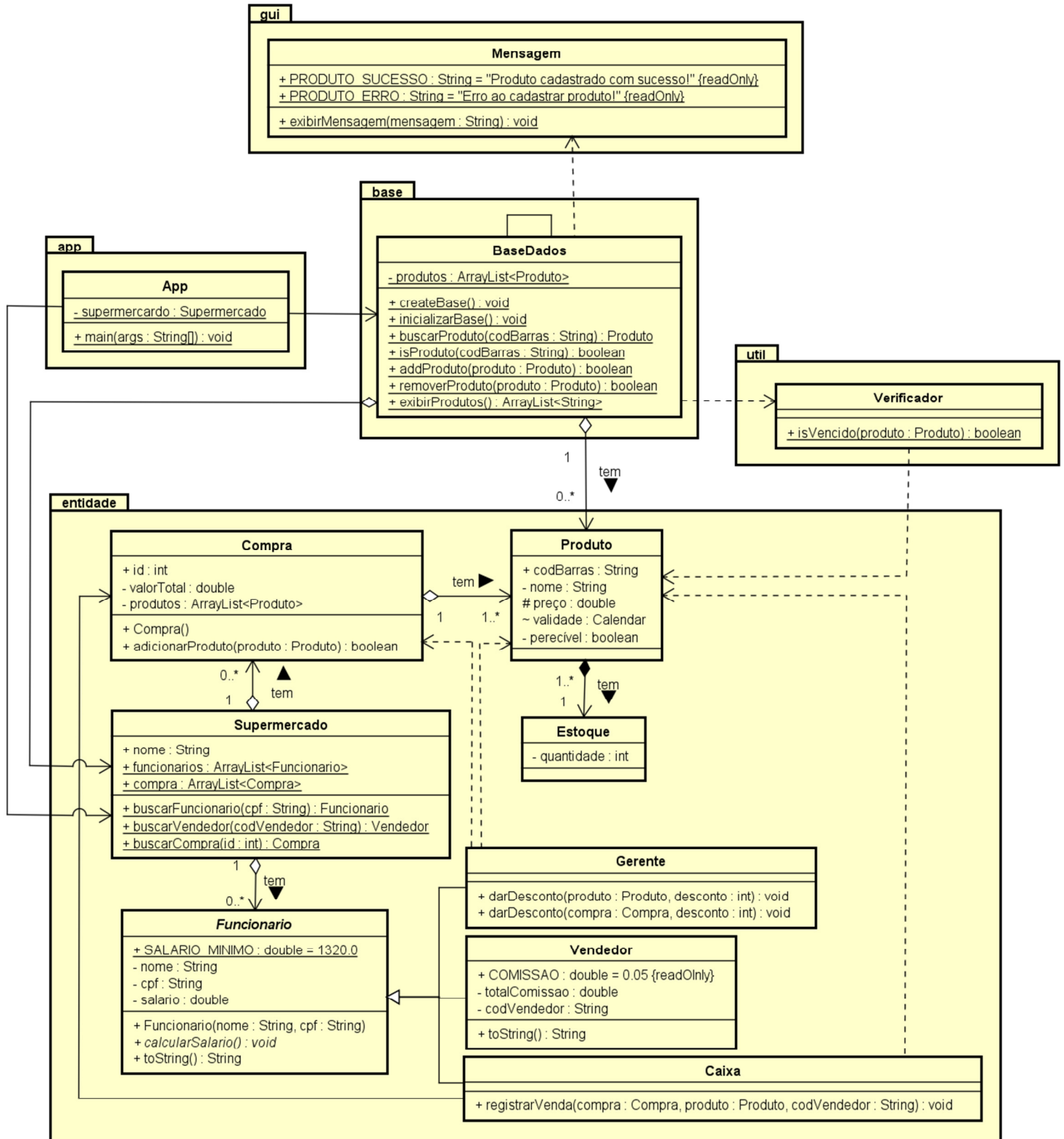
- As compras:

- A primeira com apenas um feijão realizada por João Santos.
- A segunda com 10 unidades de feijão por João Santos.
- Na terceira compra, o caixa precisa registrar apenas um arroz, mas recebeu a indicação do cliente que foi o vendedor "vend003".
- A quarta compra feita por Godofredo Sá de: 10 unidades de feijão e 10 unidades de arroz. Entretanto, o cliente solicitou um desconto de 10%.
- Na quinta compra o cliente gostaria de 10 laranjas mimosas. Mas percebeu que as laranjas estavam começando a estragar procurou o gerente para ele conceder um desconto de 50%. O gerente sabendo dessa situação passou a conceder o desconto para todo cliente. Então o caixa registrou a compra do cliente, mas sem atribuir uma comissão a um vendedor.
- Após essas compras, exiba no console:
 - Todas as compras efetuadas, mostrando no console o id e o respectivo valor.
 - As quantidades disponíveis dos produtos.
 - Os dados dos vendedores.

- 3) Durante a realização da 1ª Verificação de Aprendizagem:
- Qual assunto você teve mais dificuldade (se possível descrever com situação da prova)?
 - Qual assunto você gostaria que tivesse maior pontuação?

```
public class MP00 {
    void saudacao(){OptionPane.showMessageDialog(null, "Boa Prova!");}
    public static void main(String[] args) {
        new MP00().saudacao();
    }
}
```

APÊNDICE A



1ª VERIFICAÇÃO DE APRENDIZAGEM (PRÁTICA)

Instruções gerais:

Utilize um diretório (do pendrive) para salvar as implementações. Salve as implementações a cada modificação, caso aconteça alguma falha de energia o trabalho será preservado. Lembre-se que uma vez removido o arquivo do Eclipse, seu conteúdo será perdido.

A prova é prática e deverão ser devolvidas tanto a prova impressa quanto os códigos-fonte implementados em Java. A Nota máxima desta prova é de 10,0 pontos.

Criação e Organização de Projeto (0,25 pontos)

- 1) No Eclipse limpe todos os projetos existentes.

- Crie um novo projeto chamado

br.1va.mpoo.edu.NomeSobrenome

Este deverá ter uma pasta de pacotes chamada **supermercado** contendo todos os arquivos necessários para as respectivas questões.

- Ao finalizar a prova **compacte o projeto** contendo toda a codificação do projeto (arquivos textos e *bytecodes*) e envie-o no AVA:

[1ª Verificação de Aprendizagem] em Semana 11

- 2) O proprietário do supermercado MPOO Supermarket precisa de um sistema para seu empreendimento, como, por exemplo, gerenciar produtos, compras e funcionários. O sistema é descrito nas questões abaixo, modelado no APÊNDICE A e deverá ser implementado em Java.

Organização em pacotes (0,25 pontos)

- a) Organize seu projeto conforme os pacotes definidos.

Atributos e métodos static (0,25 pontos)

- b) Faça o devido uso de static em atributos e métodos.

Classe, atributo e método construtor (0,75 pontos)

- c) Faça a devida criação de classes, atributos e métodos construtores.

- d) É descrição do sistema (Apêndice A):

Encapsulamento e métodos de acesso (0,5 ponto)

- Observe os encapsulamento dos atributos.

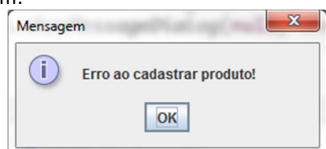
Agregação e ArrayList, definição de métodos, implementações e utilização (1,0 ponto)

- Os produtos do sistema estão armazenados na BaseDados. Faça o devido uso de ArrayList e dos métodos definidos para manipular a base:

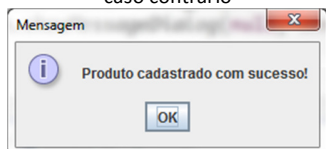
BaseDados
- produtos : ArrayList<Produto>
+ createBase() : void
+ inicializarBase() : void
+ buscarProduto(codBarras : String) : Produto
+ isProduto(codBarras : String) : boolean
+ addProduto(produto : Produto) : boolean
+ removerProduto(produto : Produto) : boolean
+ exibirProdutos() : ArrayList<String>

Nesse sistema cada produto é diferenciado pelo codBarras (único entre os produtos).

- Se um produto já cadastrado tentar ser inserido novamente ou se estiver vencido deve-se exibir uma caixa de diálogo com a mensagem:



caso contrário



- Essa verificação deverá ser realizada no método **public static boolean addProduto(Produto produto){}**, responsável por adicionar um produto no ArrayList. Esse método deverá fazer uso de outro(s) método(s), evitando a duplicidade de código.
- A validade do produto é verificada pelo serviço **isVencido** de Verificador. Portanto, não há produtos vencidos na BaseDados do sistema do supermercado.

Classes e troca de mensagens (0,25 pontos)

Fique Atento!

Você sabe como definir uma data em seu sistema? **Calendar** é classe de **java.util** que permite armazenar datas.

Mas porquê de não usar **Date** de **java.util**? A resposta é simples: *"Date is deprecated"*.

Então observe o exemplo de codificação de verificação do prazo de validade de um produto:

```
public static boolean isVencido(Produto produto){
    Calendar dataAtual = Calendar.getInstance();
    if(produto.getValidade().before(dataAtual))
        return true;
    return false;
}
```

Mas e como fazemos para criar uma data em um sistema? Por exemplo, para criar a data 11/12/2023 fazemos:

```
Calendar data = Calendar.getInstance();
data.set(2023, Calendar.DECEMBER, 11);
```

- A exibição dos produtos armazenados deverá ser por **public static ArrayList<String> exibirProduto(){}**. Mas apenas os nomes poderão ser informados.
- Implemente os demais métodos presentes na BaseDados.

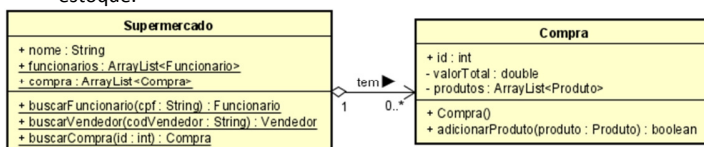
Classes e troca de mensagens (0,25 pontos)

- Todas as caixas de diálogos do sistema devem utilizar o método **exibirMensagem()** de Mensagem:

Mensagem
+ PRODUTO_SUCESSO : String = "Produto cadastrado com sucesso" (readOnly)
+ PRODUTO_ERRO : String = "Erro ao cadastrar produto" (readOnly)
+ exibirMensagem(mensagem : String) : void

Agregação e ArrayList, definição de métodos, implementações e utilização (1,0 ponto)

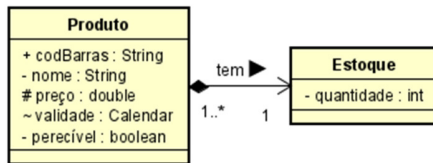
- Todos os produtos armazenados na BaseDados poderão ser comprados em uma Compra do supermercado "MPOO Supermarket", mas se houver quantidade do produto disponível em estoque.



- Todo produto adicionado em uma compra deverá ser removido do estoque (Regra de negócio definido pelo Caixa).
- Uma compra possui um identificador único que representa o número da compra efetuada, sendo este valor nunca repetido e crescente (*auto-increment*). Logo, não é permitido haver duas compras com a mesma identificação.

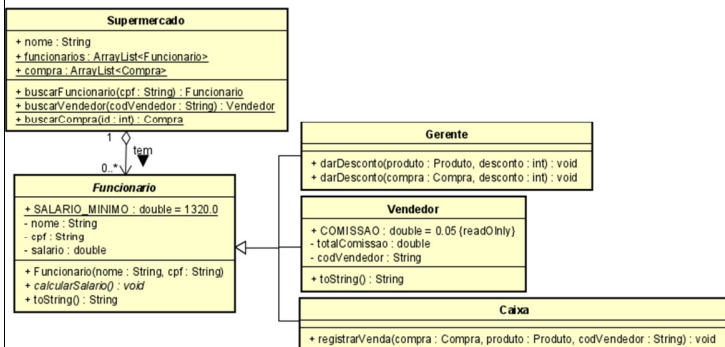
Composição (0,75 pontos)

- A relação de Produto e Estoque é de composição.

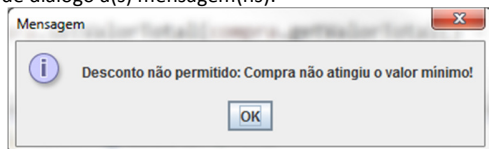


Classe e método abstract (0,5 pontos)
Herança (0,75 pontos)
Polimorfismo de Objetos (0,5 pontos)
Polimorfismo de Método (0,25 pontos)
Sobrescrita de métodos (0,25 pontos)

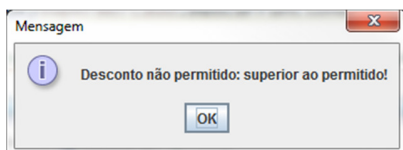
- O supermercado possui funcionários especializados do tipo Gerente, Vendedor e Caixa. Todos os funcionários são registrados em Supermercado. A manipulação dos funcionários deverá utilizar os métodos de ArrayList.



- Um gerente possui um salário de 5 salários mínimos e poderá dar descontos no valor de uma Compra ou no preço de um Produto da base. Descontos poderão ser informados apenas ao final da compra. O desconto máximo é 10% sobre o valor total de uma compra, desde que essa compra possua um valor superior a R\$ 100,00. Caso tente-se burlar o desconto, o sistema exibirá em uma caixa de diálogo a(s) mensagem(ns):



OU



- Um vendedor poderá realizar vendas de produtos. Tem salário inicial de um salário mínimo, mas que poderá ser aumentado a partir das comissões sobre produtos vendidos, para isso seu codVendedor deverá ser informado no ato do registro de uma venda.
- Um caixa recebe um salário mínimo. E tem a função de registrar os produtos de uma compra vendidos por um vendedor.
- É possível ver os dados de um funcionário por toString().

Vinculação dinâmica de método (0,5 pontos)

- O salário de um funcionário é calculado pelo método calcularSalario(). Faça o devido uso de vinculação dinâmica de método.

Documentação (0,5 pontos)

- e) Adicione o java comments para que o desenvolvedor saiba do que se trata os métodos darDesconto de Gerente.

darDesconto

```
public void darDesconto(Produto produto, int desconto)
```

Este método dar o desconto em porcentagem a um produto.

Parameters:

desconto - Desconto em porcentagem. Ex: Se preço de produto é de R\$10,00 e desconto é de 10%, então o novo preço de produto é de R\$ 9,00.

produto - Produto que receberá o desconto.

e

darDesconto

```
public void darDesconto(Compra compra, int desconto)
```

Parameters:

desconto - Desconto em porcentagem. Máximo de 10% para uma compra mínima de R\$ 100,00. Ex: Se valor de compra é de R\$100,00 e desconto é de 10%, então o novo valor da compra é de R\$ 90,00.

compra - Compra que receberá o desconto.

Instâncias e chamadas de métodos (2,0 pontos)

- f) Simule em App, a criação e utilização das instâncias:

- Produtos já cadastrados na base:

- nome: feijão da Serra, codBarras: PROD001, preço: R\$ 10,00, validade de 31/12/2024; não perecível
- nome: arroz da Serra, codBarras: PROD002, preço: R\$ 5,00, validade de 01/06/2024; não perecível
- nome: biscoito formoso, codBarras: PROD003, preço: R\$ 2,00, validade de 10/12/2023; não perecível
- nome: laranja mimosas, codBarras: PROD004, preço: R\$ 0,50, validade de 21/12/2023; perecível.

- Funcionários:

- A gerente Maria Silva (cpf 111.111.111-11) de salário R\$ 6.660,00 (cinco salários mínimo).
- O vendedor João Santos (cpf 222.222.222-22) de salário R\$ 1.320 mais comissões. Seu código: "vend001".
- O vendedor Godofredo Sá (cpf 333.333.333-33) de salário R\$ 1.320 mais comissões. Seu código: "vend002".
- O caixa José Lima (cpf 444.444.444-44) que recebe um salário mínimo.
- Exiba os dados dos funcionários do supermercado no console.

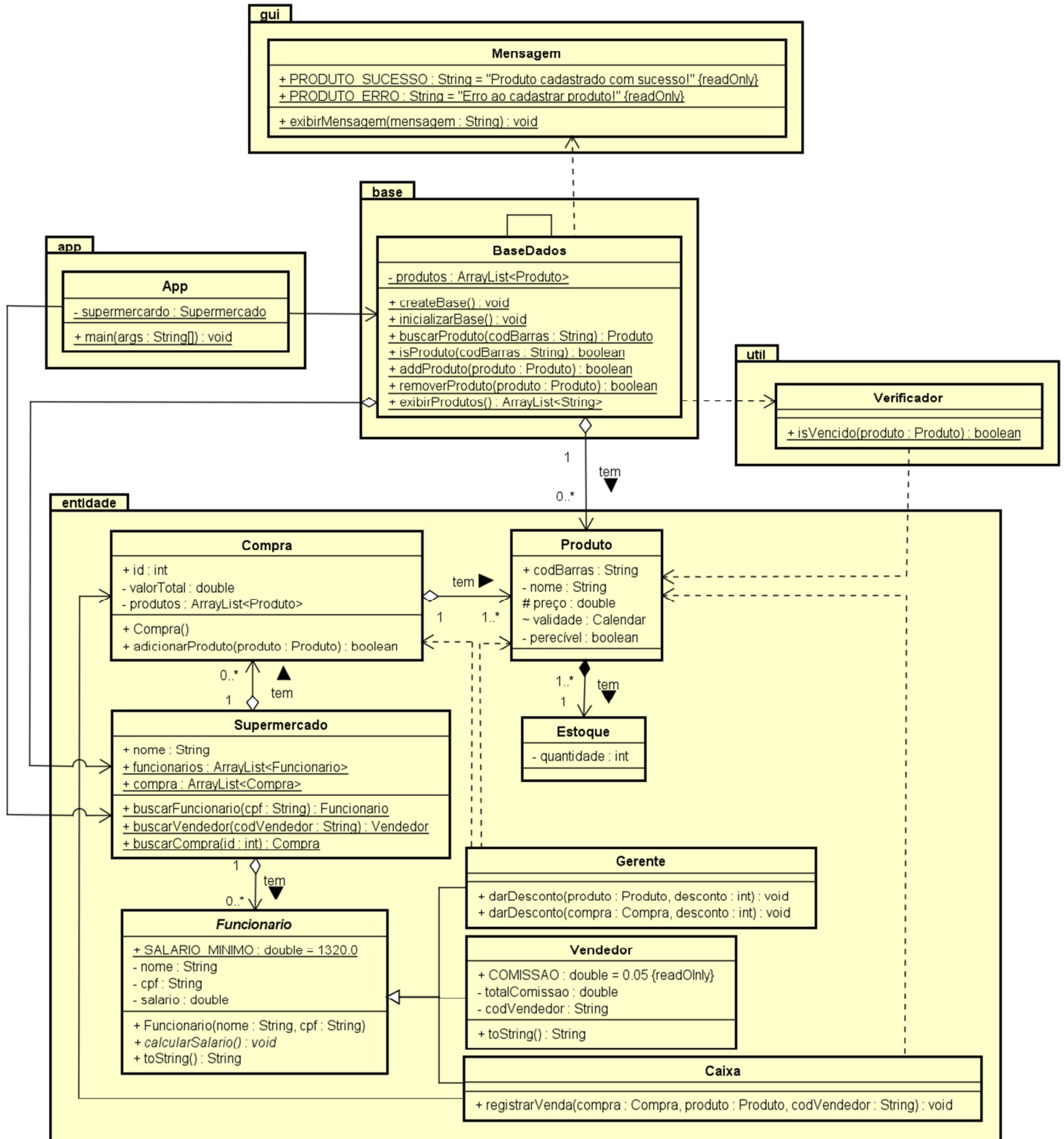
- As compras:

- A primeira com apenas um feijão realizada por João Santos.
- A segunda com 10 unidades de feijão por João Santos.
- Na terceira compra, o caixa precisa registrar apenas um arroz, mas recebeu a indicação do cliente que foi o vendedor "vend003".
- A quarta compra feita por Godofredo Sá de: 10 unidades de feijão e 10 unidades de arroz. Entretanto, o cliente solicitou um desconto de 10%.
- Na quinta compra o cliente gostaria de 10 laranjas mimosas. Mas percebeu que as laranjas estavam começando a estragar procurou o gerente para ele conceder um desconto de 50%. O gerente sabendo dessa situação passou a conceder o desconto para todo cliente. Então o caixa registrou a compra do cliente, mas sem atribuir uma comissão a um vendedor.
- Após essas compras, exiba no console:
 - Todas as compras efetuadas, mostrando no console o id e o respectivo valor.
 - As quantidades disponíveis dos produtos.
 - Os dados dos vendedores.

- 3) Durante a realização da 1ª Verificação de Aprendizagem:
- Qual assunto você teve mais dificuldade (se possível descrever com situação da prova)?
 - Qual assunto você gostaria que tivesse maior pontuação?

```
public class MP00 {
    void saudacao(){OptionPane.showMessageDialog(null, "Boa Prova!");}
    public static void main(String[] args) {
        new MP00().saudacao();
    }
}
```

APÊNDICE A



1ª VERIFICAÇÃO DE APRENDIZAGEM (PRÁTICA)

Instruções gerais:

Utilize um diretório (do pendrive) para salvar as implementações. Salve as implementações a cada modificação, caso aconteça alguma falha de energia o trabalho será preservado. Lembre-se que uma vez removido o arquivo do Eclipse, seu conteúdo será perdido.

A prova é prática e deverão ser devolvidas tanto a prova impressa quanto os códigos-fonte implementados em Java. A Nota máxima desta prova é de 10,0 pontos.

Criação e Organização de Projeto (0,25 pontos)

- 1) No Eclipse limpe todos os projetos existentes.

- Crie um novo projeto chamado

br.1va.mpoo.edu.NomeSobrenome

Este deverá ter uma pasta de pacotes chamada **sistemaSupermercado** contendo todos os arquivos necessários para as respectivas questões.

- Ao finalizar a prova **compacte o projeto** contendo toda a codificação do projeto (arquivos textos e *bytecodes*) e envie-o no AVA:

[1ª Verificação de Aprendizagem] em Semana 11

- 2) O proprietário do supermercado MPOO Supermarket precisa de um sistema para seu empreendimento, como, por exemplo, gerenciar produtos, compras e funcionários. O sistema é descrito nas questões abaixo, modelado no APÊNDICE A e deverá ser implementado em Java.

Organização em pacotes (0,25 pontos)

- a) Organize seu projeto conforme os pacotes definidos.

Atributos e métodos static (0,25 pontos)

- b) Faça o devido uso de static em atributos e métodos.

Classe, atributo e método construtor (0,75 pontos)

- c) Faça a devida criação de classes, atributos e métodos construtores.

- d) É descrição do sistema (Apêndice A):

Encapsulamento e métodos de acesso (0,5 ponto)

- Observe os encapsulamento dos atributos.

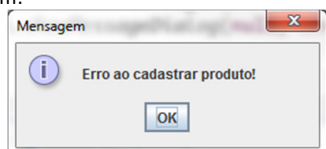
Agregação e ArrayList, definição de métodos, implementações e utilização (1,0 ponto)

- Os produtos do sistema estão armazenados na BaseDados. Faça o devido uso de ArrayList e dos métodos definidos para manipular a base:

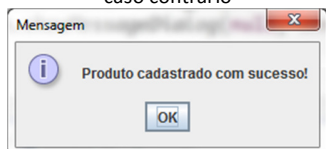
BaseDados
- produtos : ArrayList<Produto>
+ createBase() : void
+ inicializarBase() : void
+ buscarProduto(codBarras : String) : Produto
+ isProduto(codBarras : String) : boolean
+ addProduto(produto : Produto) : boolean
+ removerProduto(produto : Produto) : boolean
+ exibirProdutos() : ArrayList<String>

Nesse sistema cada produto é diferenciado pelo codBarras (único entre os produtos).

- Se um produto já cadastrado tentar ser inserido novamente ou se estiver vencido deve-se exibir uma caixa de diálogo com a mensagem:



caso contrário



- Essa verificação deverá ser realizada no método **public static boolean addProduto(Produto produto){}**, responsável por adicionar um produto no ArrayList. Esse método deverá fazer uso de outro(s) método(s), evitando a duplicidade de código.
- A validade do produto é verificada pelo serviço **isVencido** de Verificador. Portanto, não há produtos vencidos na BaseDados do sistema do supermercado.

Classes e troca de mensagens (0,25 pontos)

Fique Atento!

Você sabe como definir uma data em seu sistema? **Calendar** é classe de **java.util** que permite armazenar datas.

Mas o porquê de não usar **Date** de **java.util**? A resposta é simples: *"Date is deprecated!"*.

Então observe o exemplo de codificação de verificação do prazo de validade de um produto:

```
public static boolean isVencido(Produto produto){
    Calendar dataAtual = Calendar.getInstance();
    if(produto.getValidade().before(dataAtual))
        return true;
    return false;
}
```

Mas e como fazemos para criar uma data em um sistema? Por exemplo, para criar a data 11/12/2023 fazemos:

```
Calendar data = Calendar.getInstance();
data.set(2023, Calendar.DECEMBER, 11);
```

- A exibição dos produtos armazenados deverá ser por **public static ArrayList<String> exibirProduto(){}.** Mas apenas os nomes poderão ser informados.
- Implemente os demais métodos presentes na BaseDados.

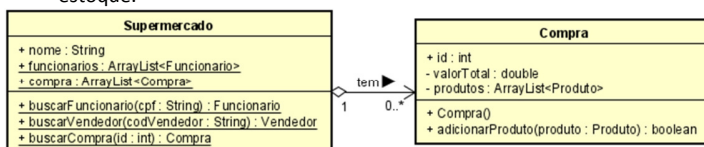
Classes e troca de mensagens (0,25 pontos)

- Todas as caixas de diálogos do sistema devem utilizar o método **exibirMensagem()** de Mensagem:

Mensagem
+ PRODUTO_SUCESSO : String = "Produto cadastrado com sucesso" (readOnly)
+ PRODUTO_ERRO : String = "Erro ao cadastrar produto" (readOnly)
+ exibirMensagem(mensagem : String) : void

Agregação e ArrayList, definição de métodos, implementações e utilização (1,0 ponto)

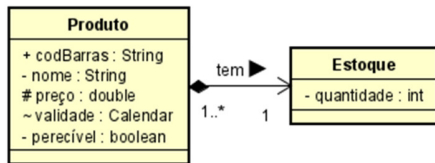
- Todos os produtos armazenados na BaseDados poderão ser comprados em uma Compra do supermercado "MPOO Supermarket", mas se houver quantidade do produto disponível em estoque.



- Todo produto adicionado em uma compra deverá ser removido do estoque (Regra de negócio definido pelo Caixa).
- Uma compra possui um identificador único que representa o número da compra efetuada, sendo este valor nunca repetido e crescente (*auto-increment*). Logo, não é permitido haver duas compras com a mesma identificação.

Composição (0,75 pontos)

- A relação de Produto e Estoque é de composição.



Classe e método abstract (0,5 pontos)

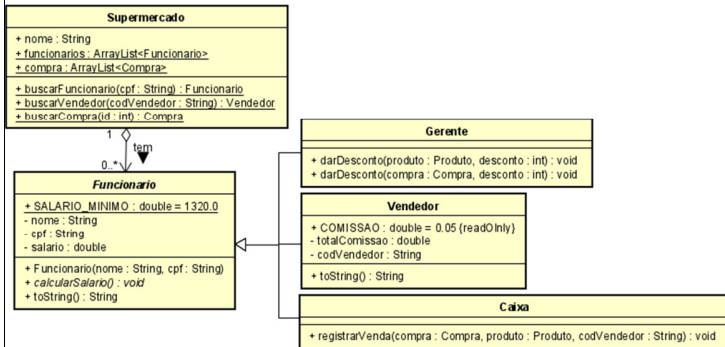
Herança (0,75 pontos)

Polimorfismo de Objetos (0,5 pontos)

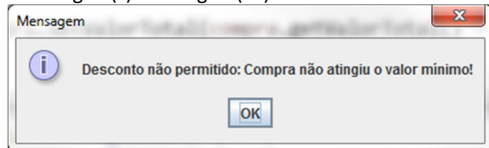
Polimorfismo de Método (0,25 pontos)

Sobrescrita de métodos (0,25 pontos)

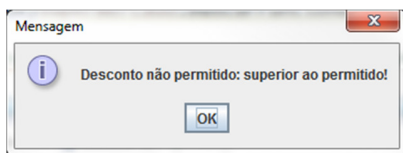
- O supermercado possui funcionários especializados do tipo Gerente, Vendedor e Caixa. Todos os funcionários são registrados em Supermercado. A manipulação dos funcionários deverá utilizar os métodos de ArrayList.



- Um gerente possui um salário de 5 salários mínimos e poderá dar descontos no valor de uma Compra ou no preço de um Produto da base. Descontos poderão ser informados apenas ao final da compra. O desconto máximo é 10% sobre o valor total de uma compra, desde que essa compra possua um valor superior a R\$ 100,00. Caso tente-se burlar o desconto, o sistema exibirá em uma caixa de diálogo a(s) mensagem(ns):



OU



- Um vendedor poderá realizar vendas de produtos. Tem salário inicial de um salário mínimo, mas que poderá ser aumentado a partir das comissões sobre produtos vendidos, para isso seu codVendedor deverá ser informado no ato do registro de uma venda.
- Um caixa recebe um salário mínimo. E tem a função de registrar os produtos de uma compra vendidos por um vendedor.
- É possível ver os dados de um funcionário por toString().

Vinculação dinâmica de método (0,5 pontos)

- O salário de um funcionário é calculado pelo método calcularSalario(). Faça o devido uso de vinculação dinâmica de método.

Documentação (0,5 pontos)

- e) Adicione o java comments para que o desenvolvedor saiba do que se trata os métodos darDesconto de Gerente.

darDesconto

```
public void darDesconto(Produto produto, int desconto)
```

Este método dar o desconto em porcentagem a um produto.

Parameters:

desconto - Desconto em porcentagem. Ex: Se preço de produto é de R\$10,00 e desconto é de 10%, então o novo preço de produto é de R\$ 9,00.

produto - Produto que receberá o desconto.

e

darDesconto

```
public void darDesconto(Compra compra, int desconto)
```

Parameters:

desconto - Desconto em porcentagem. Máximo de 10% para uma compra mínima de R\$ 100,00. Ex: Se valor de compra é de R\$100,00 e desconto é de 10%, então o novo valor da compra é de R\$ 90,00.

compra - Compra que receberá o desconto.

Instâncias e chamadas de métodos (2,0 pontos)

- f) Simule em App, a criação e utilização das instâncias:

- Produtos já cadastrados na base:

- nome: feijão da Serra, codBarras: PROD001, preço: R\$ 10,00, validade de 31/12/2024; não perecível
- nome: arroz da Serra, codBarras: PROD002, preço: R\$ 5,00, validade de 01/06/2024; não perecível
- nome: biscoito formoso, codBarras: PROD003, preço: R\$ 2,00, validade de 10/12/2023; não perecível
- nome: laranja mimosas, codBarras: PROD004, preço: R\$ 0,50, validade de 21/12/2023; perecível.

- Funcionários:

- A gerente Maria Silva (cpf 111.111.111-11) de salário R\$ 6.660,00 (cinco salários mínimo).
- O vendedor João Santos (cpf 222.222.222-22) de salário R\$ 1.320 mais comissões. Seu código: "vend001".
- O vendedor Godofredo Sá (cpf 333.333.333-33) de salário R\$ 1.320 mais comissões. Seu código: "vend002".
- O caixa José Lima (cpf 444.444.444-44) que recebe um salário mínimo.
- Exiba os dados dos funcionários do supermercado no console.

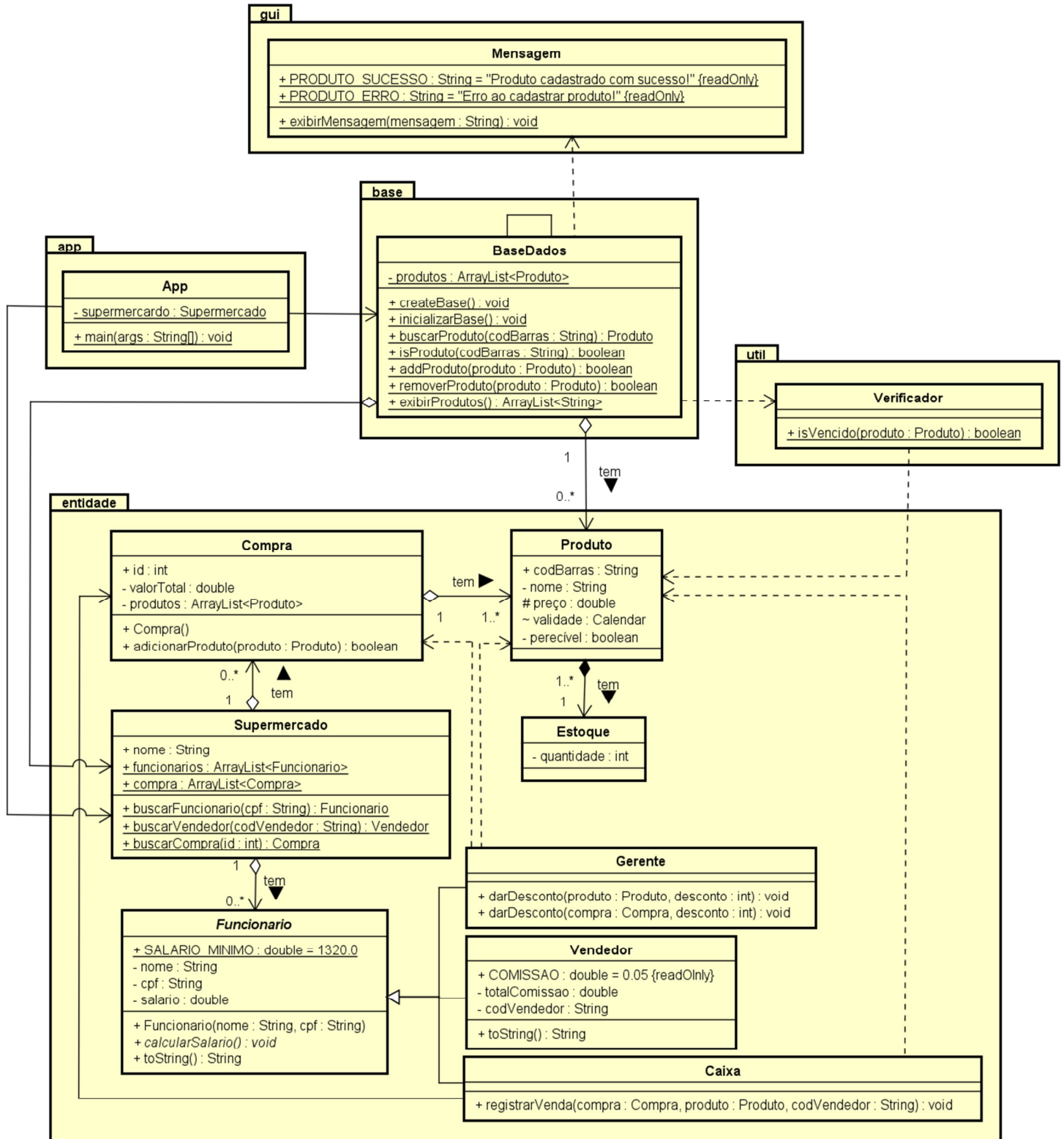
- As compras:

- A primeira com apenas um feijão realizada por João Santos.
- A segunda com 10 unidades de feijão por João Santos.
- Na terceira compra, o caixa precisa registrar apenas um arroz, mas recebeu a indicação do cliente que foi o vendedor "vend003".
- A quarta compra feita por Godofredo Sá de: 10 unidades de feijão e 10 unidades de arroz. Entretanto, o cliente solicitou um desconto de 10%.
- Na quinta compra o cliente gostaria de 10 laranjas mimosas. Mas percebeu que as laranjas estavam começando a estragar procurou o gerente para ele conceder um desconto de 50%. O gerente sabendo dessa situação passou a conceder o desconto para todo cliente. Então o caixa registrou a compra do cliente, mas sem atribuir uma comissão a um vendedor.
- Após essas compras, exiba no console:
 - Todas as compras efetuadas, mostrando no console o id e o respectivo valor.
 - As quantidades disponíveis dos produtos.
 - Os dados dos vendedores.

- 3) Durante a realização da 1ª Verificação de Aprendizagem:
- Qual assunto você teve mais dificuldade (se possível descrever com situação da prova)?
 - Qual assunto você gostaria que tivesse maior pontuação?

```
public class MP00 {
    void saudacao(){OptionPane.showMessageDialog(null, "Boa Prova!");}
    public static void main(String[] args) {
        new MP00().saudacao();
    }
}
```

APÊNDICE A



1ª VERIFICAÇÃO DE APRENDIZAGEM (PRÁTICA)

Instruções gerais:

Utilize um diretório (do pendrive) para salvar as implementações. Salve as implementações a cada modificação, caso aconteça alguma falha de energia o trabalho será preservado. Lembre-se que uma vez removido o arquivo do Eclipse, seu conteúdo será perdido.

A prova é prática e deverão ser devolvidas tanto a prova impressa quanto os códigos-fonte implementados em Java. A Nota máxima desta prova é de 10,0 pontos.

Criação e Organização de Projeto (0,25 pontos)

- 1) No Eclipse limpe todos os projetos existentes.

- Crie um novo projeto chamado

br.1va.mpoo.edu.NomeSobrenome

Este deverá ter uma pasta de pacotes chamada **supermercado** contendo todos os arquivos necessários para as respectivas questões.

- Ao finalizar a prova **compacte o projeto** contendo toda a codificação do projeto (arquivos textos e *bytecodes*) e envie-o no AVA:

[1ª Verificação de Aprendizagem] em Semana 11

- 2) O proprietário do supermercado MPOO Supermarket precisa de um sistema para seu empreendimento, como, por exemplo, gerenciar produtos, compras e funcionários. O sistema é descrito nas questões abaixo, modelado no APÊNDICE A e deverá ser implementado em Java.

Organização em pacotes (0,25 pontos)

- a) Organize seu projeto conforme os pacotes definidos.

Atributos e métodos static (0,25 pontos)

- b) Faça o devido uso de static em atributos e métodos.

Classe, atributo e método construtor (0,75 pontos)

- c) Faça a devida criação de classes, atributos e métodos construtores.

- d) É descrição do sistema (Apêndice A):

Encapsulamento e métodos de acesso (0,5 ponto)

- Observe os encapsulamento dos atributos.

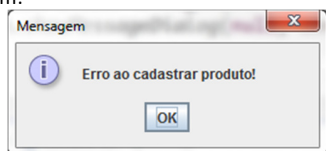
Agregação e ArrayList, definição de métodos, implementações e utilização (1,0 ponto)

- Os produtos do sistema estão armazenados na BaseDados. Faça o devido uso de ArrayList e dos métodos definidos para manipular a base:

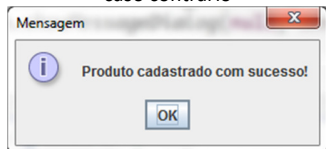
BaseDados
- produtos : ArrayList<Produto>
+ createBase() : void
+ inicializarBase() : void
+ buscarProduto(codBarras : String) : Produto
+ isProduto(codBarras : String) : boolean
+ addProduto(produto : Produto) : boolean
+ removerProduto(produto : Produto) : boolean
+ exibirProdutos() : ArrayList<String>

Nesse sistema cada produto é diferenciado pelo codBarras (único entre os produtos).

- Se um produto já cadastrado tentar ser inserido novamente ou se estiver vencido deve-se exibir uma caixa de diálogo com a mensagem:



caso contrário



- Essa verificação deverá ser realizada no método **public static boolean addProduto(Produto produto){}**, responsável por adicionar um produto no ArrayList. Esse método deverá fazer uso de outro(s) método(s), evitando a duplicidade de código.
- A validade do produto é verificada pelo serviço **isVencido** de Verificador. Portanto, não há produtos vencidos na BaseDados do sistema do supermercado.

Classes e troca de mensagens (0,25 pontos)

Fique Atento!

Você sabe como definir uma data em seu sistema? **Calendar** é classe de **java.util** que permite armazenar datas.

Mas porquê de não usar **Date** de **java.util**? A resposta é simples: *"Date is deprecated"*.

Então observe o exemplo de codificação de verificação do prazo de validade de um produto:

```
public static boolean isVencido(Produto produto){
    Calendar dataAtual = Calendar.getInstance();
    if(produto.getValidade().before(dataAtual))
        return true;
    return false;
}
```

Mas e como fazemos para criar uma data em um sistema? Por exemplo, para criar a data 11/12/2023 fazemos:

```
Calendar data = Calendar.getInstance();
data.set(2023, Calendar.DECEMBER, 11);
```

- A exibição dos produtos armazenados deverá ser por **public static ArrayList<String> exibirProduto(){}**. Mas apenas os nomes poderão ser informados.
- Implemente os demais métodos presentes na BaseDados.

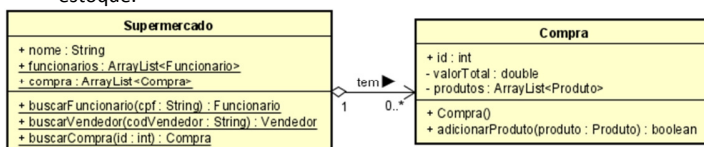
Classes e troca de mensagens (0,25 pontos)

- Todas as caixas de diálogos do sistema devem utilizar o método **exibirMensagem()** de **Mensagem**:

Mensagem
+ PRODUTO_SUCESSO : String = "Produto cadastrado com sucesso" (readOnly)
+ PRODUTO_ERRO : String = "Erro ao cadastrar produto" (readOnly)
+ exibirMensagem(mensagem : String) : void

Agregação e ArrayList, definição de métodos, implementações e utilização (1,0 ponto)

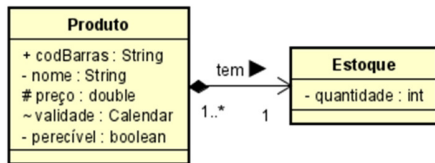
- Todos os produtos armazenados na BaseDados poderão ser comprados em uma Compra do supermercado "MPOO Supermarket", mas se houver quantidade do produto disponível em estoque.



- Todo produto adicionado em uma compra deverá ser removido do estoque (Regra de negócio definido pelo Caixa).
- Uma compra possui um identificador único que representa o número da compra efetuada, sendo este valor nunca repetido e crescente (*auto-increment*). Logo, não é permitido haver duas compras com a mesma identificação.

Composição (0,75 pontos)

- A relação de Produto e Estoque é de composição.



Classe e método abstract (0,5 pontos)

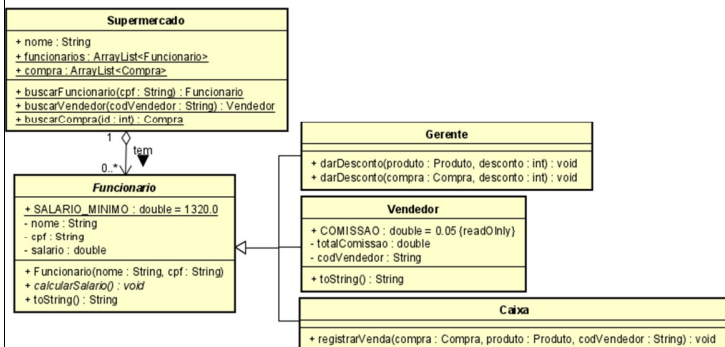
Herança (0,75 pontos)

Polimorfismo de Objetos (0,5 pontos)

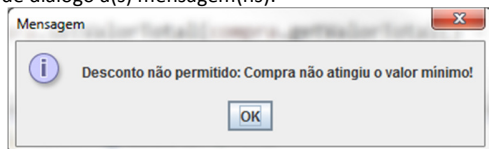
Polimorfismo de Método (0,25 pontos)

Sobrescrita de métodos (0,25 pontos)

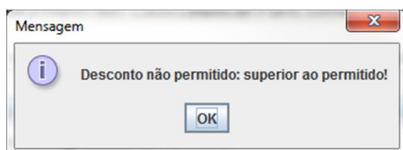
- O supermercado possui funcionários especializados do tipo Gerente, Vendedor e Caixa. Todos os funcionários são registrados em Supermercado. A manipulação dos funcionários deverá utilizar os métodos de ArrayList.



- Um gerente possui um salário de 5 salários mínimos e poderá dar descontos no valor de uma Compra ou no preço de um Produto da base. Descontos poderão ser informados apenas ao final da compra. O desconto máximo é 10% sobre o valor total de uma compra, desde que essa compra possua um valor superior a R\$ 100,00. Caso tente-se burlar o desconto, o sistema exibirá em uma caixa de diálogo a(s) mensagem(ns):



OU



- Um vendedor poderá realizar vendas de produtos. Tem salário inicial de um salário mínimo, mas que poderá ser aumentado a partir das comissões sobre produtos vendidos, para isso seu codVendedor deverá ser informado no ato do registro de uma venda.
- Um caixa recebe um salário mínimo. E tem a função de registrar os produtos de uma compra vendidos por um vendedor.
- É possível ver os dados de um funcionário por toString().

Vinculação dinâmica de método (0,5 pontos)

- O salário de um funcionário é calculado pelo método calcularSalario(). Faça o devido uso de vinculação dinâmica de método.

Documentação (0,5 pontos)

- e) Adicione o java comments para que o desenvolvedor saiba do que se trata os métodos darDesconto de Gerente.

darDesconto

```
public void darDesconto(Produto produto, int desconto)
```

Este método dar o desconto em porcentagem a um produto.

Parameters:

desconto - Desconto em porcentagem. Ex: Se preço de produto é de R\$10,00 e desconto é de 10%, então o novo preço de produto é de R\$ 9,00.

produto - Produto que receberá o desconto.

e

darDesconto

```
public void darDesconto(Compra compra, int desconto)
```

Parameters:

desconto - Desconto em porcentagem. Máximo de 10% para uma compra mínima de R\$ 100,00. Ex: Se valor de compra é de R\$100,00 e desconto é de 10%, então o novo valor da compra é de R\$ 90,00.

compra - Compra que receberá o desconto.

Instâncias e chamadas de métodos (2,0 pontos)

- f) Simule em App, a criação e utilização das instâncias:

- Produtos já cadastrados na base:

- nome: feijão da Serra, codBarras: PROD001, preço: R\$ 10,00, validade de 31/12/2024; não perecível
- nome: arroz da Serra, codBarras: PROD002, preço: R\$ 5,00, validade de 01/06/2024; não perecível
- nome: biscoito formoso, codBarras: PROD003, preço: R\$ 2,00, validade de 10/12/2023; não perecível
- nome: laranja mimosas, codBarras: PROD004, preço: R\$ 0,50, validade de 21/12/2023; perecível.

- Funcionários:

- A gerente Maria Silva (cpf 111.111.111-11) de salário R\$ 6.660,00 (cinco salários mínimo).
- O vendedor João Santos (cpf 222.222.222-22) de salário R\$ 1.320 mais comissões. Seu código: "vend001".
- O vendedor Godofredo Sá (cpf 333.333.333-33) de salário R\$ 1.320 mais comissões. Seu código: "vend002".
- O caixa José Lima (cpf 444.444.444-44) que recebe um salário mínimo.
- Exiba os dados dos funcionários do supermercado no console.

- As compras:

- A primeira com apenas um feijão realizada por João Santos.
- A segunda com 10 unidades de feijão por João Santos.
- Na terceira compra, o caixa precisa registrar apenas um arroz, mas recebeu a indicação do cliente que foi o vendedor "vend003".
- A quarta compra feita por Godofredo Sá de: 10 unidades de feijão e 10 unidades de arroz. Entretanto, o cliente solicitou um desconto de 10%.
- Na quinta compra o cliente gostaria de 10 laranjas mimosas. Mas percebeu que as laranjas estavam começando a estragar procurou o gerente para ele conceder um desconto de 50%. O gerente sabendo dessa situação passou a conceder o desconto para todo cliente. Então o caixa registrou a compra do cliente, mas sem atribuir uma comissão a um vendedor.
- Após essas compras, exiba no console:
 - Todas as compras efetuadas, mostrando no console o id e o respectivo valor.
 - As quantidades disponíveis dos produtos.
 - Os dados dos vendedores.

- 3) Durante a realização da 1ª Verificação de Aprendizagem:
- Qual assunto você teve mais dificuldade (se possível descrever com situação da prova)?
 - Qual assunto você gostaria que tivesse maior pontuação?

```
public class MP00 {
    void saudacao(){OptionPane.showMessageDialog(null, "Boa Prova!");}
    public static void main(String[] args) {
        new MP00().saudacao();
    }
}
```

APÊNDICE A

