

Pontifícia Universidade Católica do Paraná - PUCPR

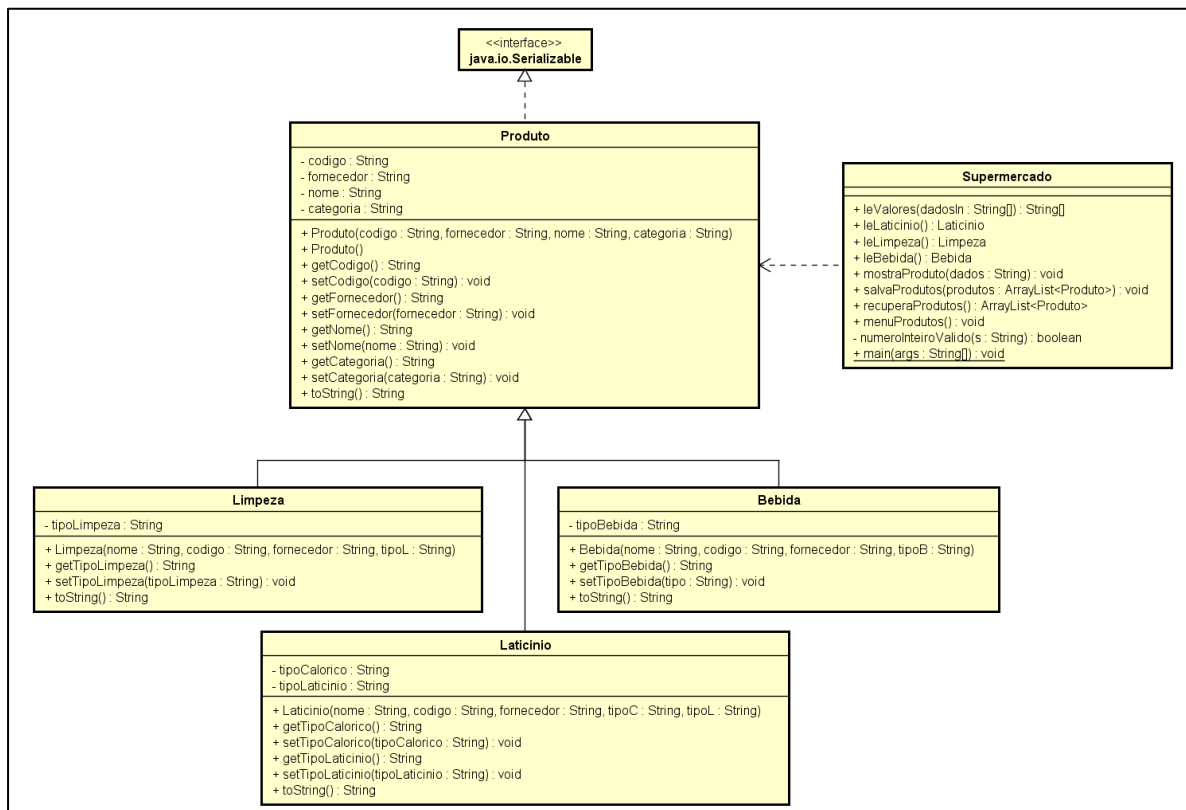
Programação Orientada a Objetos

POO - PROJETO

1. APRESENTAÇÃO

Para este trabalho, as equipes deverão construir uma aplicação em que sejam aplicados os conceitos de persistência de objetos com utilização de interface gráfica orientada a evento.

Para orientar o desenvolvimento, está disponível o exemplo (arquivo .ZIP), no qual as classes estão representadas no Diagrama de Classes a seguir.



2. QUESITOS A SEREM AVALIADOS

1. **Modelagem de classes com herança**, realizada com auxílio de ferramenta (ex. Astah);
 - a. Cada equipe deve definir sua **organização** própria de classes com herança, visando aproveitar os recursos do **polimorfismo**.
 - b. Sugestões de classes possíveis estão disponíveis no final deste documento.
2. **Polimorfismo** entre classes
 - a. **Sobrescrita (override)** de métodos
 - b. **Referências polimórficas**
 - c. Controle de **repetição** usado em coleção de objetos (**ArrayList**)
3. **Gravação e recuperação** de objetos em arquivo
4. Utilização da estrutura **switch** para controle de **menu de opções**
5. Utilização de **tratamento de exceções** para manipulação de arquivos

3. ORIENTAÇÕES

Com base no exemplo (**arquivo .ZIP**, as equipes deverão utilizar os recursos a seguir, como apresentado ou de forma equivalente, desde que todos os componentes da equipe demonstrem conhecimento desses recursos.

Lista (no exemplo, de produtos do supermercado)

- Usar estrutura de **ArrayList** para manter em memória os objetos produtos do supermercado:

```
ArrayList<Produto> produtos = new ArrayList<Produto>();
```

Menu (no exemplo, para manipulação das operações do supermercado)

- Usar estrutura de controle **switch**

```
...
switch (opcl) {
    case 1:// Entrar dados
        ...
        break;
    case 2: // Exibir dados
        ...
        break;
    case 9:
        JOptionPane.showMessageDialog
            (null,"Fim SUEPERMERCADO");
        break;
}
...
```

Sobrescrita (*override*)

- Usar no método **toString()**, aproveitando o método da classe mãe através da chamada **super**
- Usar em métodos **construtores** (encadeamento de construtores)

Persistência de objetos

- Objetos a serem persistidos (armazenados em arquivo) devem implementar a interface **java.io.Serializable**
- Gravação (**void salvaProdutos (ArrayList<Produto> produtos)**):

```
outputStream = new ObjectOutputStream
                (new FileOutputStream(nomeArquivo));
```

- Recuperação (`ArrayList<Produto> recuperaProdutos ()`):
`inputStream = new ObjectInputStream`
`(new FileInputStream(nomeArquivo)) ;`

Itens esperados na aplicação

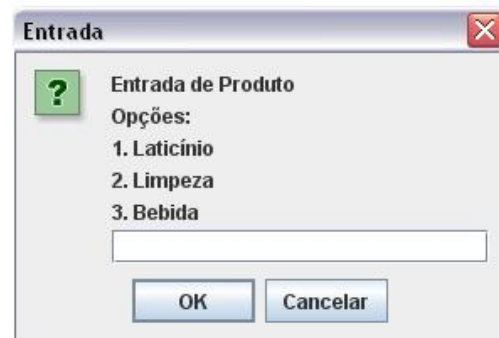
1. Cada equipe deverá ter seu **conjunto próprio de objetos (domínio de aplicação)**, objetos que deverão ser persistidos e recuperados, conforme escolha do usuário em menu de opções.
2. O conjunto de objetos deverá trabalhar a **generalização com hierarquia**, para aproveitamento das facilidades do **polimorfismo** (referência polimórfica em coleção de objetos, como **ArrayList**, para manter coleções de objetos que têm as mesmas relações hierárquicas)
3. A aplicação deverá utilizar **interface gráfica**, conforme exemplificado (supermercado.zip)
4. A aplicação deverá ter **tratamento de exceção** para acesso a arquivos e para tratamento de demais erros em potencial
5. A aplicação deverá ter tratamento de **opções para o usuário via menu**.
6. Utilizar **construtores** adequadamente (reutilizar da classe mãe, sempre que possível)
7. Utilizar **getter** e **setter** para prover encapsulamento, sempre que possível
8. Utilizar o método `toString()` para apresentar objetos ao usuário.

Menus esperados (como no exemplo, para manipulação das operações do supermercado)

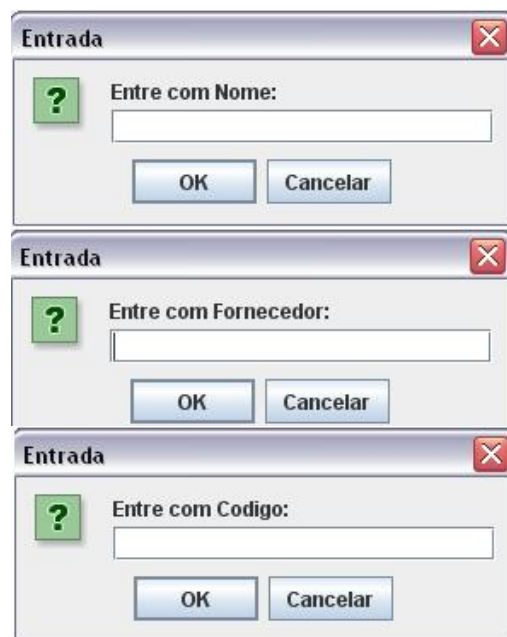
- Opções da aplicação




- Opção 1) Entrada de dados geral:



- Opção 1) Entrada de dados para todos os tipos de produtos:



- Opção 1) Complementação - entrada de dados para Laticínios:



The image shows two screenshots of a dialog box titled "Entrada". The first screenshot shows the dialog with the label "Entre com Tipo Laticínio:" and an empty text input field. The second screenshot shows the dialog with the label "Entre com Tipo Calorico:" and an empty text input field. Both dialogs have "OK" and "Cancelar" buttons at the bottom.

- Opção 1) Complementação - entrada de dados para Limpeza:



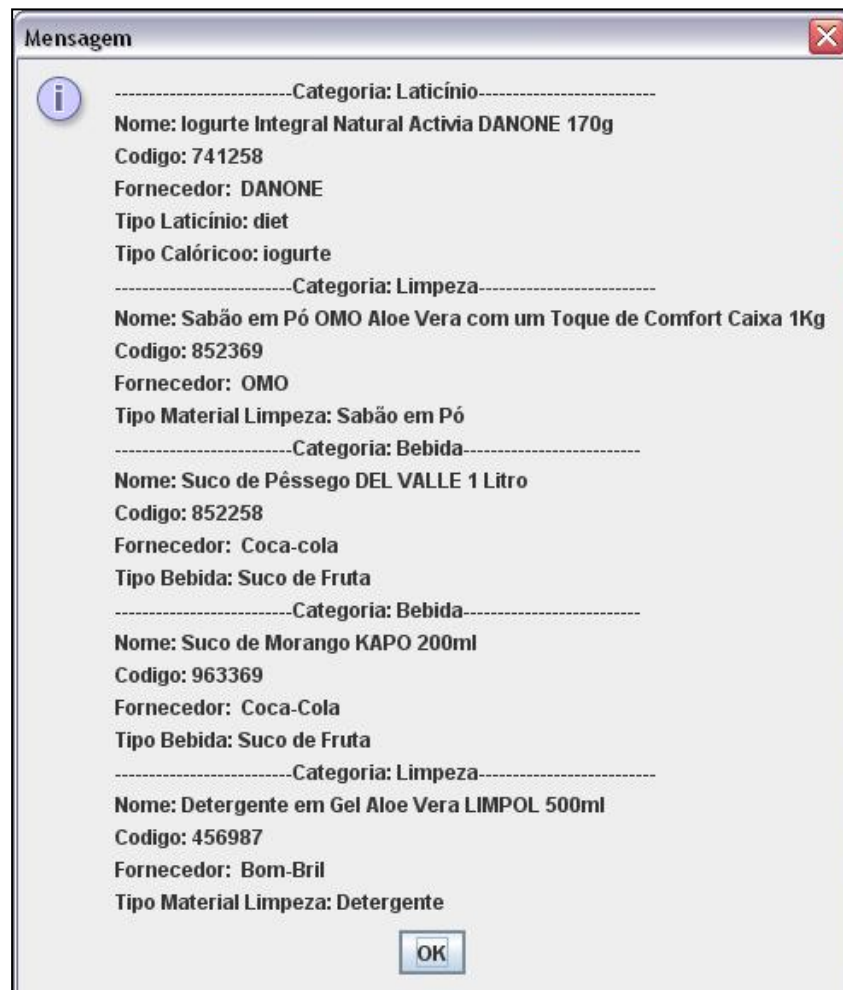
The image shows a screenshot of a dialog box titled "Entrada". It contains the label "Entre com Tipo Limpeza:" and an empty text input field. At the bottom, there are "OK" and "Cancelar" buttons.

- Opção 1) Complementação - entrada de dados para Bebidas:

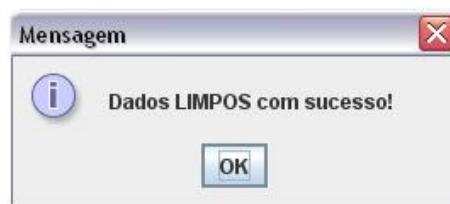


The image shows a screenshot of a dialog box titled "Entrada". It contains the label "Entre com Tipo Bebida:" and an empty text input field. At the bottom, there are "OK" and "Cancelar" buttons.

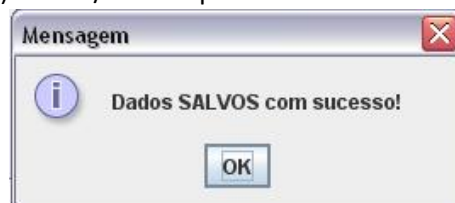
- Opção 2) Exibir dados:



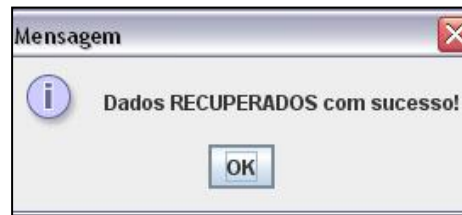
- Opção 3) Limpar estrutura de memória que mantém produtos



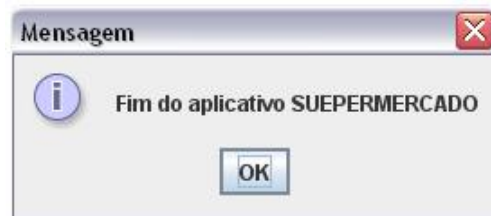
- Opção 4) Salvar / Gravar produtos



- Opção 5) Recuperar / Ler produtos



- Opção 9) Sair da aplicação



- Verificação para as opções “2) Exibir dados” e “4) Salvar dados”:



3. SUGESTÕES DE DOMÍNIO DE APLICAÇÃO

Seguem sugestões para alteração do domínio de aplicação, para que as equipes alterem o exemplo passado. Caso prefiram, as equipes poderão criar e utilizar seu próprio conjunto de objetos, desde que validado com o professor.

Biblioteca

- Livro (nome, ISBN, autor, resenha)
 - Livro Infantis (brinde)
 - Livros de Culinária (tipo de culinária)
 - Guias de Viagem (local)

Clínica Veterinária

- Animal (nome, espécie, dono)
 - Cavalo (alimentação e cuidados próprios)
 - Cachorro (alimentação e cuidados próprios)
 - Pássaro (alimentação e cuidados próprios)

Exames Clínicos

- Exame (convênio, nome médico, nome paciente)
 - Sangue (preparação própria)
 - Raio X (preparação própria)
 - Ecografia (preparação própria)

Grupo Escolar

- Estudante (matrícula, nome, responsável)
 - Pré-escolar (cuidados)
 - Ensino fundamental (atividades de contra turno)
 - Ensino médio (oficinas profissionalizantes)

Frota Motorizada

- Veículo (marca, modelo, ano, quilometragem, placa)
 - Automóvel (motorização)
 - Caminhão (carga máxima)
 - Ônibus (assentos)

Delegação

- Atleta (nome, número)
 - Saltador (altura)
 - Corredor (velocidade)
 - Nadador (estilo)

Loja de eletrônicos

- Equipamento (nome, marca, modelo, tamanho da tela)
 - Smartphone (quantidade de chips de operadora, tamanho da tela)
 - Smartwatch (tipo pulseira)
 - Notebook (processador, memória RAM, disco, tamanho da tela)