

FIAP

TURMA 1TDSPV / 1TDSPZ

DOMAIN DRIVEN DESIGN USING JAVA

PROFESSOR: TIAGO MENDES FERRER

GLOBAL SOLUTION - ECONOMIA AZUL -2024

PAULO ANDRÉ CARMINATI

RM 557881 / 1TDSPZ

ARTHUR BISPO DE LIMA

RM 557868 / 1TDSPV

JOÃO PAULO MOREIRA DOS SANTOS

RM 557808 / 1TDSPV

SUMÁRIO

CAPA	1
SUMÁRIO	2
DESCRIPTIVO	3
DIAGRAMA DE CLASSES.....	5
OBSERVAÇÕES.....	7
DIAGRAMA DE CLASSES DESENHO	8

DESCRITIVO

O seguinte diagrama de classes é uma versão prévia que será desenvolvida em Java. De início, serve para suprir as demandas de entregas exigidas pela faculdade referente a avaliação de nota.

Objetivo atual de sprint 2 é demonstrar as classes bean e suas ligações com outras classes e atributos de referência, assim, como herança estendidas e interfaces implementadas via métodos feitos com valores de entrada criados pelo usuário e demonstrados nas classes de execução, com algum tratamento de erro inicial. Criou-se ainda onde necessário os métodos toString que podem se desejar ser implementados na classe de execução.

Em resumo, no momento o projeto conta com 6 classes, sendo 1 superclasse. E 3 interfaces implementando métodos em outras 3 classes.

Como justificativa, temos o pedido de desenvolvimento de um sistema por parte da Faculdade para a Global solution Economia Azul.

Em nosso projeto imaginamos um jogo onde crianças de escolas podem participar trazendo materiais recicláveis que seriam reconhecidos e pesados a partir de um aparelho feito em Arduino, com isso, por exemplo, cada tipo de material pesado, catalogado e devidamente classificado geraria pontos e prêmios em um jogo interativo. Assim, ensinando a importância da reciclagem e de se proteger os mares e os oceanos dos problemas dos plásticos, jogados nestes locais sem a devida reciclagem.

O diagrama de classes atual e inicial comporta atualmente as seguintes classes:

- 1. Alunos;**
- 2. Escolas;**
- 3. Enderecos (superclasse);**
- 4. Materiais;**
- 5. Jogos;**

- 6. Iescolas (interface);**
- 7. Imateriais (interface);**
- 8. Ijogos (interface);**

DIAGRAMA DE CLASSES

1 – Classe Alunos:

Cadastra os dados dos alunos participantes e recebe como referência: escolas, materiais e jogos.

2 – Classe Escolas:

Cadastra os dados das escolas a qual os alunos pertencem e recebe como referência: alunos.

3 – Classe Enderecos:

Comporta os atributos descritos no diagrama e trabalha como superclasses de herança de Alunos e Escolas

6 – Classe Materiais:

Comporta atributos próprios para gerar uma lista de matérias entregues pelos alunos e sua pesagem.

7 – Classe Jogos:

Comporta os atributos próprios e gera atributos para serem guardados pontos e gerar lista de pontos para os alunos.

9 – Interface Iescola:

Implementa um método `exibeEscola` que exibe o nome e tipo da escola concatenados.

10 – Interface Imateriais:

Implementa um método `totalMateriais` que exibe o total de materiais entregues.

11 – Interface Imateriais:

Implementa um método `calculaPontos` que exibe os pontos cadastrados.

OBSERVAÇÃO:

Para que CERTOSDETALHES não fiquem repetitivos, explico-os neste momento:

- 1- Todas as classes recebem um método construtor de classe vazio;
- 2- Todas as classes recebem um método construtor cheio;
- 3- Todas as classes têm a geração dos métodos get e set.
- 4- Todas as classes com exceção de Endereços, receberam um método ToString, para futuro uso se necessário.
- 5- Foi feito uma classe de execução ao estilo menu com apresentação para que se possa fazer um melhor teste das funcionalidades vias System.out .
- 6- Todas as classes em específico possuem um atributo código que posteriormente será autogerado pelo SGBD, e já pensando nele. Desta forma, preferimos não gerar seus métodos get e set, assim como, dentro do construtor cheio, para não sermos obrigados em caso de criação na classe teste de entrada de dados via construtor deste campo. Por isso, ele foi criado, mas não com tais métodos. E para não ficar estranho a todo o momento na classe de execução o usuário digitar o código da classe.

DIAGRAMA DE CLASSES DESENHO

