A baixo segue explicação das classes e funcionamento delas

1. **Classe** MuscleGroup

**Papel:** Representa um grupo muscular no corpo humano.

**Atributos:**

* name: (String) O nome do grupo muscular.

**Métodos:**

* Construtor: Inicializa a classe com o nome do grupo muscular.
* getName(): Retorna o nome do grupo muscular.

**Explicação:** Essa classe encapsula informações sobre os grupos musculares. No programa, ela serve para armazenar nomes como "Peitoral", "Abdômen" ou "Glúteos", que serão exibidos nas telas.

2. **Classe** Exercise

**Papel:** Representa um exercício específico, com uma descrição associada.

**Atributos:**

* name: (String) Nome do exercício.
* description: (String) Uma descrição detalhada do exercício (como as etapas para executá-lo).

**Métodos:**

* Construtor: Inicializa a classe com o nome e a descrição do exercício.
* getExerciseDetails(): Retorna uma string com o nome do exercício e sua descrição.

**Explicação:** Essa classe encapsula informações de um exercício, permitindo que sejam exibidas nas telas de exercícios. Por exemplo, o exercício "Abdominal" pode ter instruções como "Deite no chão...".

3. **Classe Abstrata** MapScreen

**Papel:** Representa uma tela base (genérica) usada como modelo para outras telas no aplicativo.

**Atributos:**

* title: (String) Título da tela.

**Métodos:**

* Construtor: Inicializa o título da tela.
* displayScreen(): Um método abstrato que será sobrescrito pelas subclasses.

**Explicação:** Essa classe define a estrutura comum para todas as telas do programa, permitindo que as subclasses reutilizem o título e implementem comportamentos específicos. O uso de herança evita duplicação de código.

4. **Classe** GenderMapScreen **(Subclasse de** MapScreen**)**

**Papel:** Representa uma tela de mapa corporal, mostrando grupos musculares específicos, dependendo do gênero.

**Atributos:**

* muscleGroups: (Array de MuscleGroup) Lista dos grupos musculares exibidos na tela.

**Métodos:**

* Construtor: Inicializa a tela com um título e os grupos musculares.
* displayScreen(): Implementa a exibição da tela, mostrando o título e os nomes dos grupos musculares.

**Explicação:** Essa classe herda de MapScreen e especializa o método displayScreen() para exibir os grupos musculares. É usada para criar diferentes telas para gêneros masculinos e femininos.

5. **Classe** ExerciseScreen **(Subclasse de** MapScreen**)**

**Papel:** Representa uma tela que exibe os detalhes de um exercício específico.

**Atributos:**

* exercise: (Objeto da classe Exercise) O exercício que será exibido na tela.

**Métodos:**

* Construtor: Inicializa a tela com um título e um exercício.
* displayScreen(): Implementa a exibição da tela, mostrando o título e os detalhes do exercício.

**Explicação:** Assim como GenderMapScreen, essa classe também herda de MapScreen. Ela especializa o método displayScreen() para exibir as informações de um exercício específico.

6. **Classe** GymHelpApp

**Papel:** Classe principal que inicializa e executa o aplicativo.

**Funções Importantes:**

* Instancia objetos das classes MuscleGroup, Exercise, GenderMapScreen e ExerciseScreen.
* Exibe as telas utilizando os métodos displayScreen().

**Explicação:** Essa classe é o ponto de entrada do programa. Ela cria os dados necessários (grupos musculares e exercícios), instancia as telas específicas e invoca os métodos para exibir as informações. É aqui que o usuário "navega" pelas telas virtualmente.

Relação entre as Classes

* **Herança:** MapScreen é a classe base para GenderMapScreen e ExerciseScreen, promovendo reutilização de código.
* **Encapsulamento:** As propriedades (name, description, etc.) são privadas e acessadas por métodos públicos (getters).
* **Polimorfismo:** O método displayScreen() é sobrescrito pelas subclasses para se comportar de maneira diferente em GenderMapScreen e ExerciseScreen.

Exemplo de Funcionamento

1. A classe principal GymHelpApp cria grupos musculares (MuscleGroup) e exercícios (Exercise).
2. Com base nesses dados, instancias de GenderMapScreen e ExerciseScreen são criadas.
3. A tela é exibida chamando o método displayScreen() das subclasses, com comportamentos apropriados para cada tela.