

Aula prática nº 8

Tópicos

- Tipos genéricos
- Classes abstratas
- Interfaces
- Problemas POO

Exercícios

1. A classe `java.util.ArrayList<E>` é uma implementação de um vetor dinâmico fornecida pelo JavaSE. Consulte brevemente a documentação desta classe e execute o seguinte programa.

```
public class ALDemo {

    public static void main(String[] args) {
        List<Integer> c1 = new ArrayList<>();
        for (int i = 10; i <= 100; i+=10)
            c1.add(i);
        System.out.println("Size: " + c1.size());
        for (int i = 0; i < c1.size(); i++)
            System.out.println("Elemento: " + c1.get(i));

        List<String> c2 = new ArrayList<>();
        c2.add("Chuva");
        c2.add("Frio");
        c2.add("Vento");
        System.out.println("Size: " + c2.size());
        for (int i = 0; i < c2.size(); i++)
            System.out.println("Elemento: " + c2.get(i));
        c2.remove(0);
        c2.remove(0);
        for (int i = 0; i < c2.size(); i++)
            System.out.println("Elemento: " + c2.get(i));
    }
}
```

- a) Modifique-o livremente para testar algumas das funções que estão disponíveis nesta classe.
 - b) Experimente utilizar este exemplo com classes desenvolvidas em aulas anteriores (Figura, Pessoa, etc.).
2. Retome a solução que desenvolveu para o problema 7.1. Construa agora uma nova classe `Conjunto<E>` que permita criar instâncias do tipo:

```
Conjunto<Figura> desenhos = new Conjunto<>()
Conjunto<Pessoa> multidaos = new Conjunto<>()
Conjunto<Integer> medidas = new Conjunto<>()
```

Teste a implementação com o mesmo código *main* acrescentando outras instâncias, como os exemplos anteriores.

3. Considere as seguintes entidades e características representativas de alimentos:

- Carne, tem variedade (vaca, porco, peru, frango, outra), proteínas (double), calorias (double), peso (double).
- Peixe, tem tipo (congelado ou fresco), proteínas (double), calorias (double), peso (double).
- Cereal, tem nome (String), proteínas (double), calorias (double), peso (double). É um alimento vegetariano.
- Legume, tem nome (String), proteínas (double), calorias (double), peso (double). É um alimento vegetariano.
- Prato, tem um nome (String) e composição (conjunto de alimentos)
- PratoVegetariano, tem um nome (String) e composição (conjunto de alimentos vegetarianos).
- PratoDieta, tem um nome (String) e composição (conjunto de alimentos) e limite máximo de calorias (double).
- Ementa – tem um nome (String), um local (String) e uma lista de pratos associados a cada dia da semana. Este devem ser mantidos ordenados por dia.

a) Analise o problema cuidadosamente e modele as interfaces e classes necessárias, as suas associações (herança, composição) bem como todos os atributos e métodos. Implemente todas as classes necessária, seguindo as seguintes considerações:

- Os valores de calorias e proteínas usados devem ser relativos a 100gr.
- Para cada prato deve ser possível obter informações sobre alimentos, peso total, calorias, proteínas,...
- Implemente os métodos `hashCode()`, `equals()`, `toString()` em todas as classes.
- Os pratos devem respeitar a interface `Comparable` para permitir ordenação (a ordenação será por calorias).

b) Teste a implementação com o seguinte programa:

```
public class Test {

    public static void main(String[] args) {
        Ementa ementa = new Ementa("Especial Caloiro", "Snack da UA");
        Prato[] pratos = new Prato[10];

        for (int i=0; i < pratos.length; i++){
            pratos[i] = randPrato(i);
            int cnt = 0;

            while (cnt < 2){ // Adicionar 2 Ingredientes a cada Prato
                Alimento aux = randAlimento();
                if (pratos[i].addIngrediente(aux))
                    cnt++;
                else
                    System.out.println("ERRO: Não é possível adicionar '" +
                                        aux + "' ao -> " + pratos[i]);
            }
            ementa.addPrato(pratos[i], DiaSemana.rand()); // Dia Aleatório
        }
        System.out.println("\n" + ementa);
    }
}
```

```
// Retorna um Alimento Aleatoriamente
public static Alimento randAlimento() {
    switch ((int) (Math.random() * 4)) {
        default:
        case 0:
            return new Carne(VariedadeCarne.fringo, 22.3, 345.3, 300);
        case 1:
            return new Peixe(TipoPeixe.congelado, 31.3, 25.3, 200);
        case 2:
            return new Legume("Couve Flor", 21.3, 22.4, 150);
        case 3:
            return new Cereal("Milho", 19.3, 32.4, 110);
    }
}

// Retorna um Tipo de Prato Aleatoriamente
public static Prato randPrato(int i) {
    switch ((int) (Math.random() * 3)) {
        default:
        case 0:
            return new Prato("Prato N." + i);
        case 1:
            return new PratoVegetariano("Prato N." + i + " (Vegetariano)");
        case 2:
            return new PratoDieta("Prato N." + i + " (Dieta)", 90.8);
    }
}
}
```

Verifique se obteve um resultado similar a este:

```
ERRO: Não é possível adicionar 'Carne frango, Proteínas 22.3, calorias 345.3, Peso 300.0' ao -
> Dieta (0.0 Calorias) Prato 'Prato N.0 (Dieta)' composto por 0 Ingredientes
ERRO: Não é possível adicionar 'Carne frango, Proteínas 22.3, calorias 345.3, Peso 300.0' ao -
> Vegetariano Prato 'Prato N.3 (Vegetariano)' composto por 0 Ingredientes
ERRO: Não é possível adicionar 'Peixe congelado, Proteínas 31.3, calorias 25.3, Peso 200.0' ao -
-> Vegetariano Prato 'Prato N.3 (Vegetariano)' composto por 0 Ingredientes
ERRO: Não é possível adicionar 'Peixe congelado, Proteínas 31.3, calorias 25.3, Peso 200.0' ao -
-> Vegetariano Prato 'Prato N.3 (Vegetariano)' composto por 1 Ingredientes
ERRO: Não é possível adicionar 'Peixe congelado, Proteínas 31.3, calorias 25.3, Peso 200.0' ao -
-> Vegetariano Prato 'Prato N.3 (Vegetariano)' composto por 1 Ingredientes
ERRO: Não é possível adicionar 'Carne frango, Proteínas 22.3, calorias 345.3, Peso 300.0' ao -
> Vegetariano Prato 'Prato N.3 (Vegetariano)' composto por 1 Ingredientes
ERRO: Não é possível adicionar 'Carne frango, Proteínas 22.3, calorias 345.3, Peso 300.0' ao -
> Vegetariano Prato 'Prato N.4 (Vegetariano)' composto por 1 Ingredientes
ERRO: Não é possível adicionar 'Carne frango, Proteínas 22.3, calorias 345.3, Peso 300.0' ao -
> Vegetariano Prato 'Prato N.4 (Vegetariano)' composto por 1 Ingredientes
ERRO: Não é possível adicionar 'Peixe congelado, Proteínas 31.3, calorias 25.3, Peso 200.0' ao -
-> Vegetariano Prato 'Prato N.5 (Vegetariano)' composto por 1 Ingredientes
ERRO: Não é possível adicionar 'Carne frango, Proteínas 22.3, calorias 345.3, Peso 300.0' ao -
> Dieta (0.0 Calorias) Prato 'Prato N.9 (Dieta)' composto por 0 Ingredientes
Dieta (84.2 Calorias) Prato 'Prato N.0 (Dieta)' composto por 2 Ingredientes, dia Segunda
Prato 'Prato N.1' composto por 2 Ingredientes, dia Terça
Prato 'Prato N.7' composto por 2 Ingredientes, dia Terça
Vegetariano Prato 'Prato N.2 (Vegetariano)' composto por 2 Ingredientes, dia Quarta
Prato 'Prato N.8' composto por 2 Ingredientes, dia Quarta
Vegetariano Prato 'Prato N.3 (Vegetariano)' composto por 2 Ingredientes, dia Quinta
Vegetariano Prato 'Prato N.5 (Vegetariano)' composto por 2 Ingredientes, dia Quinta
Dieta (84.2 Calorias) Prato 'Prato N.6 (Dieta)' composto por 2 Ingredientes, dia Quinta
Dieta (71.28 Calorias) Prato 'Prato N.9 (Dieta)' composto por 2 Ingredientes, dia Sexta
Vegetariano Prato 'Prato N.4 (Vegetariano)' composto por 2 Ingredientes, dia Domingo
```

c) Construa um programa Ementa que permita, genericamente:

- Ingrediente
 - Adicionar Carne
 - Adicionar Peixe
 - Adicionar Cereal
 - Adicionar Legume
- Prato
 - Cria Prato
 - Apaga Prato
 - Seleciona Prato
 - Adiciona Ingrediente
 - Remove Ingrediente
- Ementa
 - Adiciona Prato
 - Remove Prato
 - Imprime Ementa