PROGRAMAÇÃO I - CCT0827

Semana Aula: 15

Unidade 4 - Coleções

Tema

Coleções

Palavras-chave

Coleção, ArrayList, Wrapper, Generics

Objetivos

O aluno deverá ser capaz de:

- Compreender o uso de Generics
- Realizar aplicações com ArrayList

Estrutura de Conteúdo

Generics

Como vimos nos exemplos anteriores, quando recuperamos um objeto de uma coleção temos que usar o *cast*. Isso ocorre porque, na sua versão original, essas coleções trabalham com o tipo Object, e os métodos que retornam dados das coleções também retornam o tipo Object.

Trabalhar com o tipo Object, nesses caso, tem duas desvantagens:

1. Tem que sempre usar o *cast* para retornar ao tipo original que foi armazenado na coleção;

2. Possibilita a inserção de objetos "errados" nas coleções (por exemplo, o programador pode se enganar e inserir um objeto Turma em uma lista de Alunos).

O conceito de Generics veio para resolver esse problema. Podemos definir o tipo a ser armazenado nas coleções colocando o nome do tipo desejado entre < e > logo após o nome da coleção:

ArrayList<**Aluno**> alunos;

ArrayList<**Float**> notas;

ArrayList<**String**> palavras;

ArrayList<**Veiculo**> veiculos; //é possível armazenar carro ou caminhão nesta lista

ArrayList<**Conta**> contas;

Para criar a coleção colocamos os símbolos <> logo após o nome da coleção:

alunos = new ArrayList<>();

notas = new ArrayList<>();

palavras = new ArrayList<>();

veículos = new ArrayList<>();

contas = new ArrayList<>();

Importante: Podemos retirar o <> que o código compilará normalmente, mas o compilador Java emitirá alguns avisos.

- Se tentarmos armazenar objetos de tipos diferentes daquele declarado, ocorrerá um erro de compilação.
- Ao recuperar um objeto da coleção não precisamos mais do *casting*, pois o compilador já sabe que tipo está armazenado na coleção.

Exemplo:

ArrayList<**Aluno**> alunos = new ArrayList<>();

alunos.add(new Turma(1, "POO")); //Erro de compilação : tipos incompatíveis

Aluno a = alunos.get(5); // Não precisamos do **cast**, pois o método get() retorna **Aluno** !

Estratégias de Aprendizagem

Para que o aprendizado seja proveitoso, o aluno deve se envolver ativamente na participação da aula, deve realizar as tarefas propostas, realizar testes por conta própria nos programas desenvolvidos e compartilhar sua experiência/conclusão com todos.

Toda tarefa realizada pode ser conferida com o professor, para que haja certeza se está ou não correta.

Indicação de Leitura Específica

Aplicação: articulação teoria e prática

Exercício : Faça um programa em Java que crie uma lista de alunos, sabendo que qualquer aluno possui nome, matrícula e média. Depois de criada a lista , faça o que se pede:

a) imprima todos os dados de todos os alunos.

b) imprima os nomes do alunos com média abaixo de 6.0.

Considerações Adicionais