

Dado o problema abaixo, implemente uma solução em uma linguagem orientada a objetos utilizando padrões de projeto conforme sugerido no enunciado do exercício. Após finalizar, encaminhe o projeto completo compactado para a turma virtual da disciplina no Google Classroom.

1 Descrição do problema:

Faça a implementação de uma lista duplamente encadeada de números inteiros. Sua implementação deve seguir o padrão de projetos Iterator, conforme diagrama de classes da figura 1.

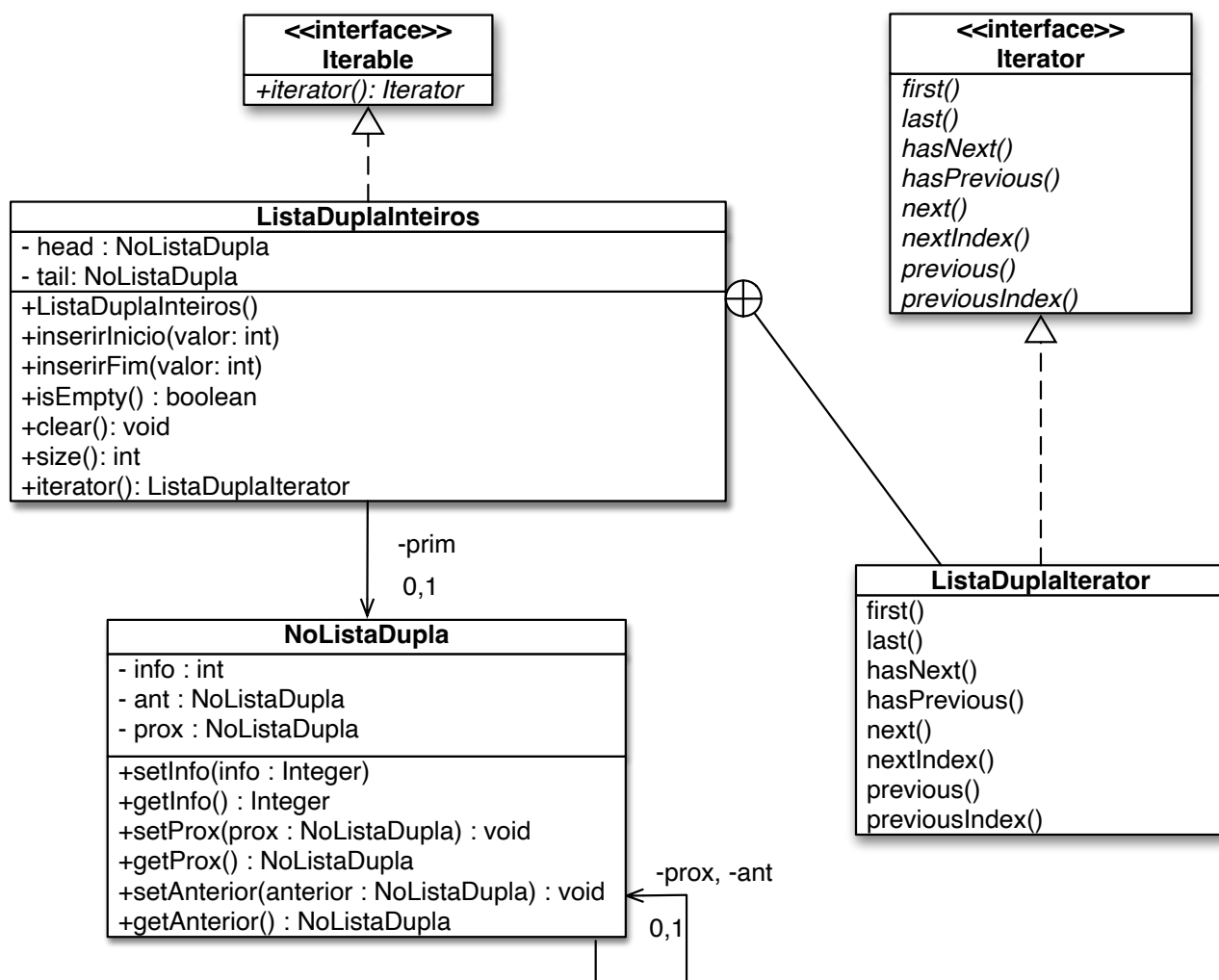


Figura 1: diagrama de classes da implementação da lista com iterador

Observações:

- Implemente as três classes (e todos os métodos) do diagrama da figura 1;
- Implemente um exemplo de programa principal demonstrando o uso de sua lista com iterador.

2 Descrição das classes e métodos:

2.1 Classe ListaDuplaInteiros

Esta classe implementa a interface **Iterable**, portanto seu método **iterator** deve instanciar e retornar um iterador concreto da lista dupla. Os atributos **head** e **tail** são referências para o primeiro e último nó da lista, respectivamente. A lista é inicializada vazia com ambos os atributos com valor nulo.

2.2 Classe ListaDuplaIterator

- **first()**: deve mover o “cursor” ou posição do iterador para o primeiro nó da lista. Se a lista estiver vazia, este cursor deve ter valor nulo;
- **last()**: deve mover o “cursor” ou posição do iterador para o último nó da lista. Se a lista estiver vazia, este cursor deve ter valor nulo;
- **hasNext()**: deve retornar um valor booleano indicando se ainda há elemento da lista a ser iterado (“visitado” pelo iterador) no sentido início→fim ;
- **hasPrevious()**: deve retornar um valor booleano indicando se ainda há elemento da lista a ser iterado (“visitado” pelo iterador) no sentido reverso, ou seja no sentido fim→início;
- **next()**: retorna o valor (inteiro) do nó atual e move o cursor para o próximo nó;
- **nextIndex()**: retorna o índice do próximo elemento do objeto iterador. Se não houver um próximo elemento, ele retorna o tamanho da lista subjacente;
- **previous()**: retorna o valor (inteiro) do nó atual e move o cursor para o nó anterior da lista subjacente;
- **previousIndex()**: retorna o índice do elemento anterior do objeto iterador. Se não houver um elemento anterior, ele retorna -1.