

Lab IX.

Objetivos

Os objetivos deste trabalho são:

- Utilizar padrões estruturais (i.e., *Chain of Responsibility*, *Command*, *Interpreter*, *Iterator*) para resolver casos práticos.
- Aplicar boas práticas de programação por padrões

Nota: Para além do código no codeUA, inclua também um ficheiro PDF ou PNG com o diagrama de classes da solução final.

IX.1 Implementação de iteradores sobre um conjunto

Considere o código seguinte:

```
public class VectorGeneric<T> {
    private T[] vec;
    private int nElem;
    private final static int ALLOC = 50;
    private int dimVec = ALLOC;

    @SuppressWarnings("unchecked")
    public VectorGeneric() {
        vec = (T[]) new Object[dimVec];
        nElem = 0;
    }

    public boolean addElem(T elem) {
        if (elem == null)
            return false;
        ensureSpace();
        vec[nElem++] = elem;
        return true;
    }

    private void ensureSpace() {
        if (nElem >= dimVec) {
            dimVec += ALLOC;
            @SuppressWarnings("unchecked")
            T[] newArray = (T[]) new Object[dimVec];
            System.arraycopy(vec, 0, newArray, 0, nElem);
            vec = newArray;
        }
    }

    public boolean removeElem(T elem) {
        for (int i = 0; i < nElem; i++) {
            if (vec[i].equals(elem)) {
                if (nElem - i - 1 > 0) // not last element
                    System.arraycopy(vec, i + 1, vec, i, nElem - i - 1);
                vec[--nElem] = null; // libertar último objecto para o GC
                return true;
            }
        }
        return false;
    }
}
```

```

    public int totalElem() {
        return nElem;
    }

    public T getElem(int i) {
        return (T) vec[i];
    }
}

```

- a) Construa o código necessário para que a classe passe a incluir os seguintes métodos:

```

public java.util.Iterator<E> Iterator()
public java.util.ListIterator<E> listIterator()
public java.util.ListIterator<E> listIterator(index) // start at index

```

Não implemente os métodos opcionais e respeite rigorosamente os contratos especificados na documentação `java8`.

- b) Desenvolva uma classe de teste para verificar todas as operações criadas. Inclua a situação de usar vários iteradores em simultâneo sobre o mesmo conjunto.

IX.2 Command

Usando o padrão *Command*, construa uma classe para **adicionar um elemento** a uma coleção (`java.util.Collection<E>`) permitindo realizar a operação *undo*. Repita a metodologia para uma classe que remova um elemento de uma coleção (com possibilidade de *undo*).

IX.3 Chain of Responsibility

Reveja o padrão *Chain of Responsibility* e proponha/construa um exemplo de aplicação que não repita a solução apresentada nos slides.