

Started on Saturday, 15 October 2022, 5:14 PM**State** Finished**Completed on** Saturday, 15 October 2022, 5:24 PM**Time taken** 9 mins 48 secs**Marks** 3,00/5,00**Grade** 12,00 out of 20,00 (60%)**Question 1**

Incorrect

Mark 0,00 out of 1,00

Considera a assinatura da classe StackArrayList

```
public class StackArrayList<T> implements Stack<T>
```

Qual a instrução correta para criar uma instancia `s1` do tipo Stack com elementos do tipo `Disco` ?

Select one:

- ☒ a. `Stack<Disco> s1 = new Stack<Disco>();` ✖
- ☐ b. `Stack s1 = new StackArrayList(Disco d);`
- ☐ c. `Stack<Disco> s1 = new StackArrayList();`
- ☐ d. `Stack s1 ; s1 = (Disco) new StackArrayList();`

Question 2

Incorrect

Mark 0,00 out of 1,00

Na implementação de ADTs na Linguagem Java a representação concreta dos dados é feita:

Select one:

- ☒ a. Nos ficheiros de dados ✖
- ☐ b. Nos métodos das classes
- ☐ c. Nos atributos das classes
- ☐ d. Nas Interfaces

Question 3

Correct

Mark 1,00 out of 1,00

Pretende-se definir no ADT Queue que a operação **enqueue**, esta deve dar um erro caso a fila esteja cheia. Qual a especificação mais correta para o método **enqueue**?

Select one:

- ☐ a. `int enqueue(E elem)`
- ☒ b. `void enqueue(E elem) throws QueueFullException` ✓
- ☐ c. `void enqueue(E elem) throws Exception`
- ☐ d. `void enqueue(E elem)`

Question 4

Correct

Mark 1,00 out of 1,00

Considere o seguinte excerto de código, e indique qual a estrutura de dados que é utilizada na implementação da ADT Stack.

```
public class S<T> implements Stack<T> {  
    private T[] elements;  
    private int size;
```

Select one:

- ☐ a. Array de Objectos
- ☒ b. ArrayList ✓
- ☐ c. Nenhuma das anteriores
- ☐ d. LinkedList com nós duplos

Question 5

Correct

Mark 1,00 out of 1,00

Considere o seguinte excerto de código, relativo à implementação da operação `pop()` usando como estrutura base o `ArrayList`. Qual dos snippets completa corretamente o código em falta ?

```
public class StackArray<E> implements Stack<E> {  
    private E[] storage;  
    private int size;  
    private final static int MAX = 500;  
  
    public StackArray(int capacity) {  
        storage = (E[]) new Object[capacity];  
        size = 0;  
    }  
    public E peek() throws EmptyStackException {  
        if (isEmpty()) {  
            throw new EmptyStackException();  
        }  
        return storage[size - 1];  
    }  
  
    @Override  
    public E pop() throws EmptyStackException {  
        // Snippet??  
  
        return elem;  
    }  
  
    //Snippet1  
    if (isEmpty()) throw new EmptyStackException();  
    E elem = storage[size - 1];  
    storage[size--] = null;  
  
    //Snippet2  
    E elem = storage[--size];  
    storage[size+1] = null;  
  
    //Snippet3  
    if (isEmpty()) throw new EmptyStackException();  
    E elem = storage[size - 1];  
    storage[--size] = null;
```

Select one:

- ☐ a. Snippet1
- ☐ b. Nenhum dos anteriores
- ☐ c. Snippet2
- ☒ d. Snippet3 ✓