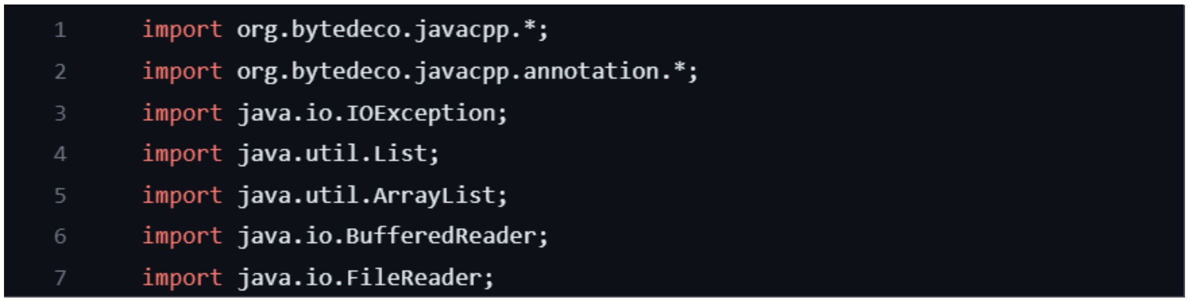
Documentation st0244-2023-2-lab1 Interoperabilty Java and C++

Team: Felipe Castro Jaimes and Juan Esteban Zuluaga

This code of Java and C++ demonstrates how to load shared library using JavaCPP, executing a function from a C++ file in Java. The most important goal of this program is to find the maximum number in a list of numbers that is provided in a text file. We are going to explain each part of the code and its duty.

Java Code

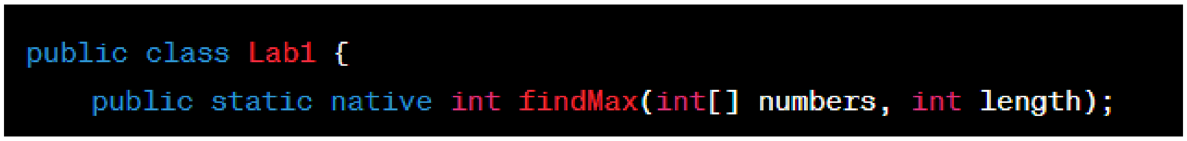
- *Imports*



```java
import org.bytedeco.javacpp.*;
import org.bytedeco.javacpp.annotation.*;
import java.io.IOException;
import java.util.List;
import java.util.ArrayList;
import java.io.BufferedReader;
import java.io.FileReader;
```

In this section, we import the necessary libraries that we are going to use in the program from JavaCPP and other standard Java classes and utilities.

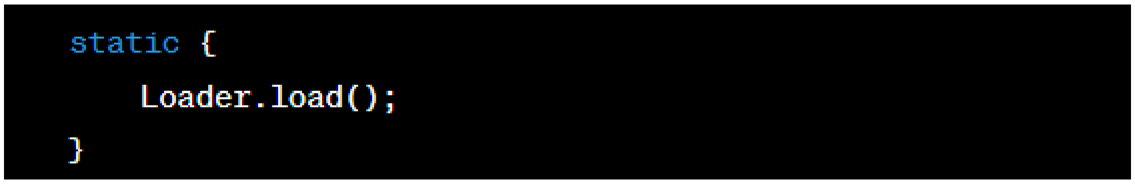
- *Declaration of Native Function*



```java
public class Lab1 {
    public static native int findMax(int[] numbers, int length);
```

In this part, we declare a native function called findMax, which will be implemented in the C++ file. This function will be used to find the maximum number in a list of integers.

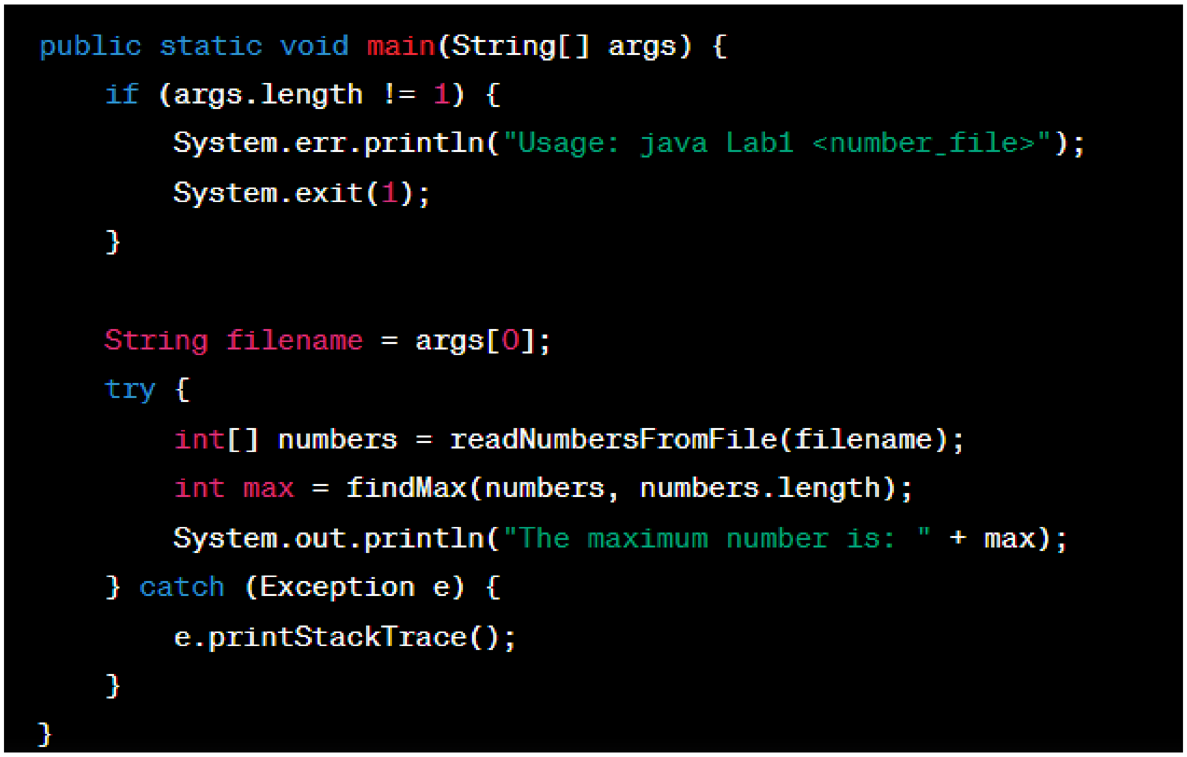
- *Loading the Shared library*



```java
  static {
      Loader.load();
  }
```

Here, we use the static method, Loader.load() to load the shared library containing the implementation of the findMax function. The library is loaded when the Lab1 class is loaded.

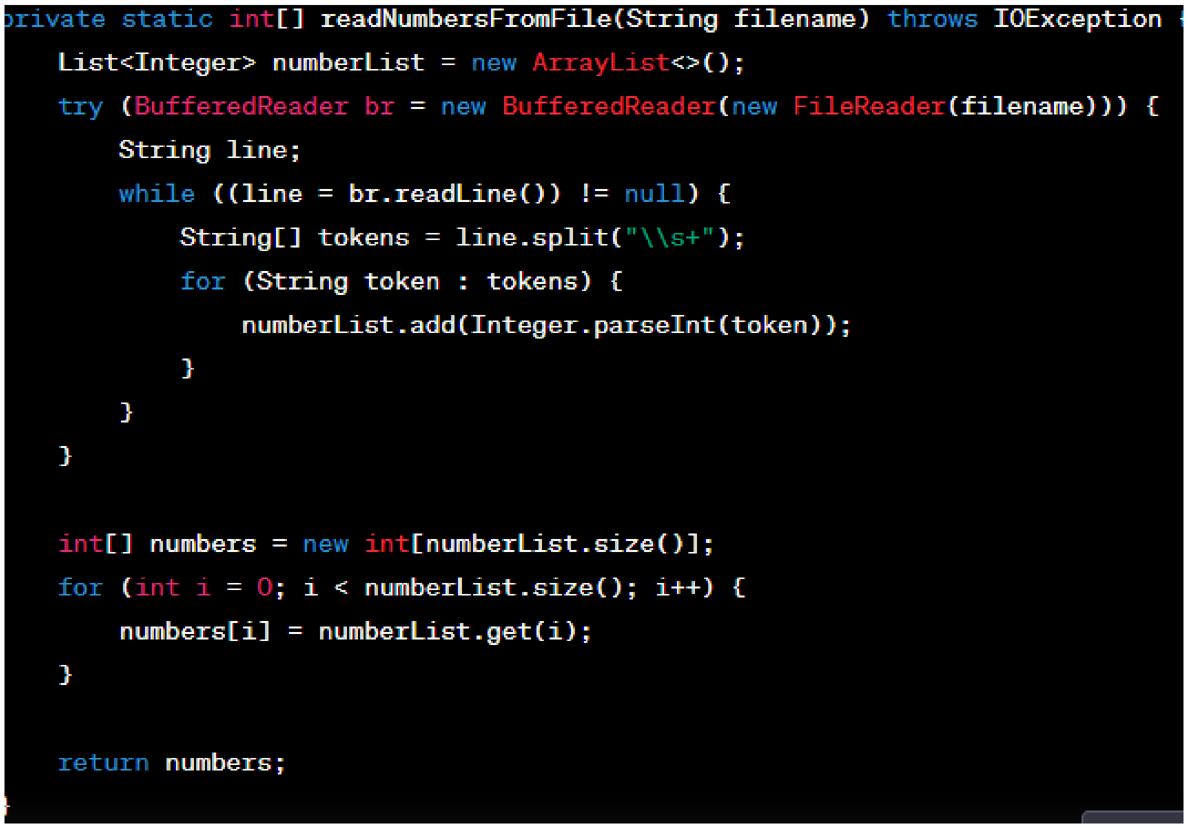
- *Main Function (main)*



```
public static void main(String[] args) {
    if (args.length != 1) {
        System.err.println("Usage: java Lab1 <number_file>");
        System.exit(1);
    }

    String filename = args[0];
    try {
        int[] numbers = readNumbersFromFile(filename);
        int max = findMax(numbers, numbers.length);
        System.out.println("The maximum number is: " + max);
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

In the main method, we check whether a command-line argument specifying the filename containing the numbers has been provided. Then, we read the file's contents and call the native function findMax from C++ file to find the maximum number. The result is printed to the standard console.

- *Reading Numbers from a File*

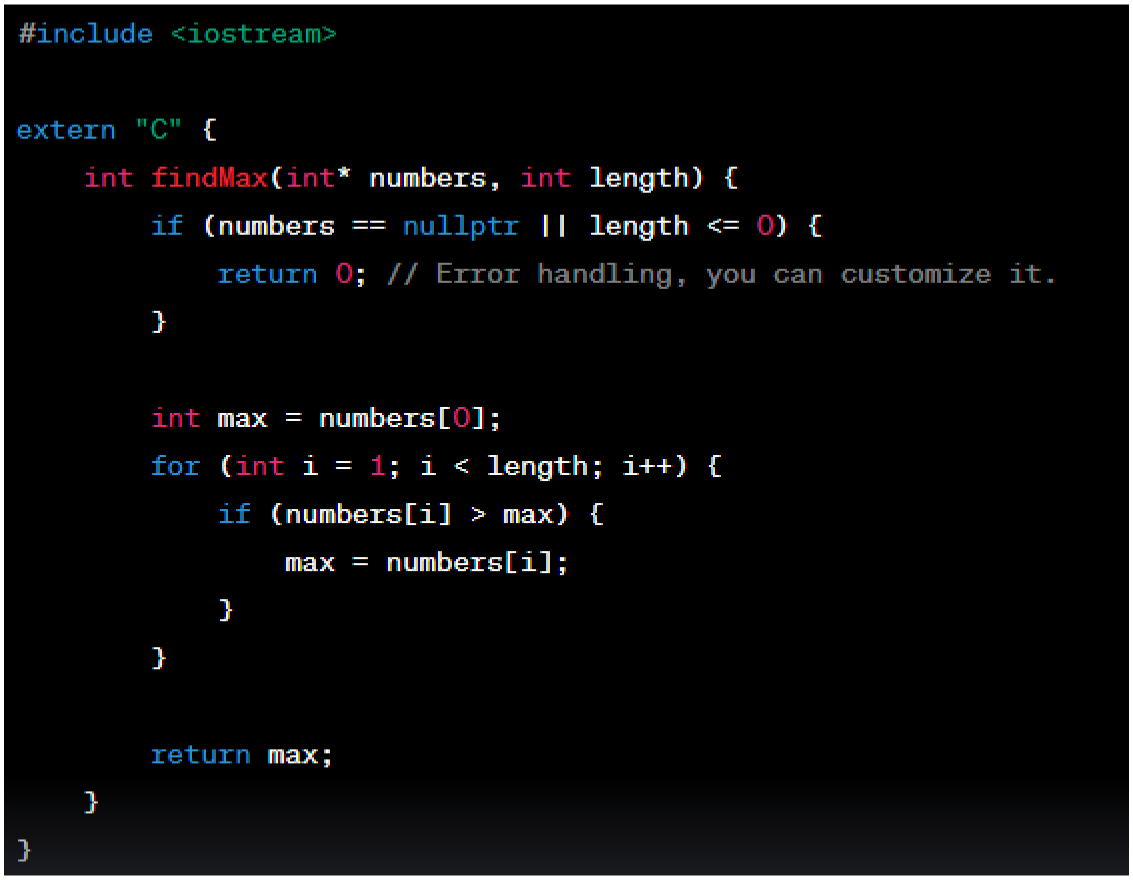


```
rivate static int[] readNumbersFromFile(String filename) throws IOException
    List<Integer> numberList = new ArrayList<>();
    try (BufferedReader br = new BufferedReader(new FileReader(filename))) {
        String line;
        while ((line = br.readLine()) != null) {
            String[] tokens = line.split("\\s+");
            for (String token : tokens) {
                numberList.add(Integer.parseInt(token));
            }
        }
    }

    int[] numbers = new int[numberList.size()];
    for (int i = 0; i < numberList.size(); i++) {
        numbers[i] = numberList.get(i);
    }

    return numbers;
}
```

This private function, readNumbersFromFile, is responsible of the duty of reading numbers from the specified file and storing them in an integer array. We use a BufferedReader to read the file line by line and then split each line into tokens to extract the numbers. The numbers are stored in a dynamic list numberList, which is later converted into an integer array numbers and returned.

C++ Code

- *Implementation of the findMax Function*



```cpp
#include <iostream>

extern "C" {
    int findMax(int* numbers, int length) {
        if (numbers == nullptr || length <= 0) {
            return 0; // Error handling, you can customize it.
        }

        int max = numbers[0];
        for (int i = 1; i < length; i++) {
            if (numbers[i] > max) {
                max = numbers[i];
            }
        }

        return max;
    }
}
```

In the C++ file, we implement the findMax function. This function takes an array of integers and its length as arguments and returns the maximum number in the array. We check whether the numbers pointer is null or if the length is less than or equal to zero for error handling.

Clarifications

We as a team would like to clarify that, through the process of developing the program we experimented some limitations that affected the velocity of developing because of these issues. We could show the teacher those errors and issues that were in the program with the aim of hoping to find a solution. But at the end we would like to thank the teacher for his help and make him a request. That if the teacher can find a way to delete or correct those errors, we would like to ask him to keep us up to the date. This with the aim of learning and acquiring more knowledge.

Conclusion

This Java and C++ code demonstrates how to integrate C++ code into a Java application using JavaCPP. The native function findMax implemented in C++ is loaded and used in Java to find the maximum number in a list of numbers provided in a text file. This documentation provides an overview of each part of the code and how they relate to achieve the final goal.