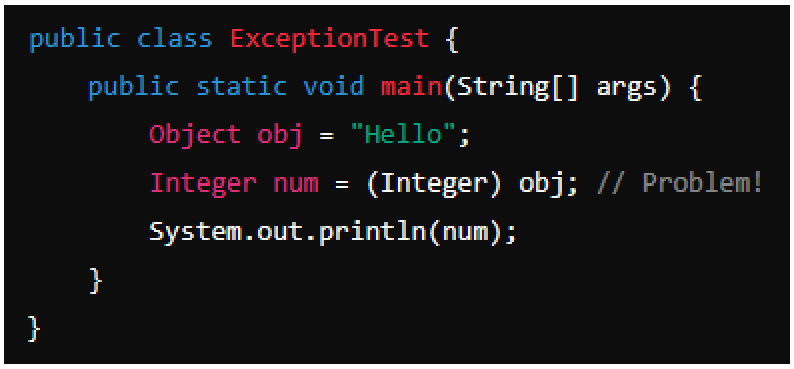
20250211 Pflichtaufgabe

Aufgabe 1

Try to predict the exception, and write a short explanation.

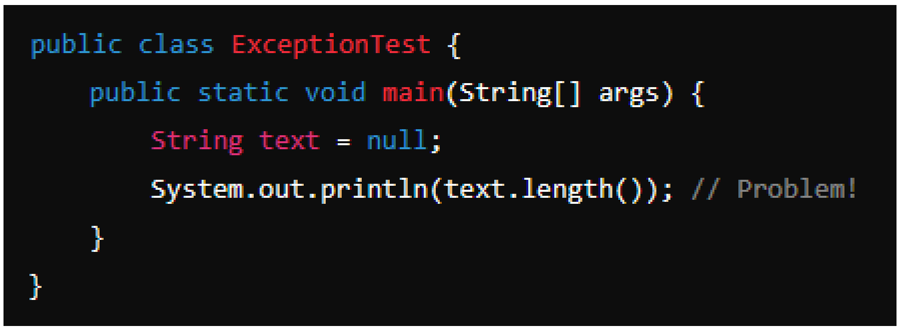
1.1



```java
public class ExceptionTest {
    public static void main(String[] args) {
        Object obj = "Hello";
        Integer num = (Integer) obj; // Problem!
        System.out.println(num);
    }
}
```

What exception is thrown and why?

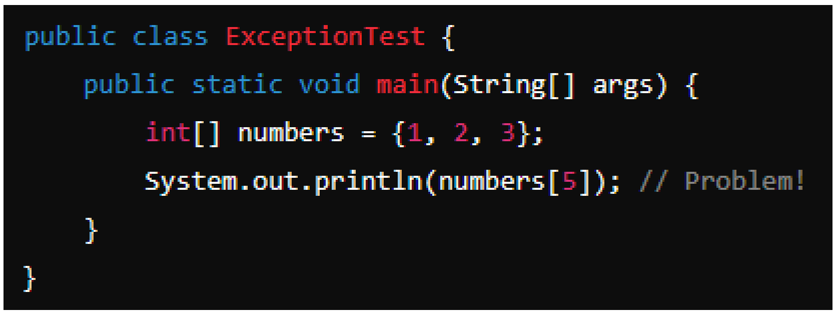
1.2



```java
public class ExceptionTest {
    public static void main(String[] args) {
        String text = null;
        System.out.println(text.length()); // Problem!
    }
}
```

What exception is thrown and why?

1.3



```java
public class ExceptionTest {
    public static void main(String[] args) {
        int[] numbers = {1, 2, 3};
        System.out.println(numbers[5]); // Problem!
    }
}
```

What exception is thrown and why?

1.4

```java
public class ExceptionTest {
    public static void setAge(int age) {
        if (age < 0) {
            throw new IllegalArgumentException("Age cannot be negative!");
        }
        System.out.println("Age is: " + age);
    }

    public static void main(String[] args) {
        setAge(-5); // Problem!
    }
}
```

What exception is thrown and why?

1.5

```java
public class ExceptionTest {
    public static void main(String[] args) {
        String number = "123abc";
        int result = Integer.parseInt(number); // Problem!
        System.out.println(result);
    }
}
```

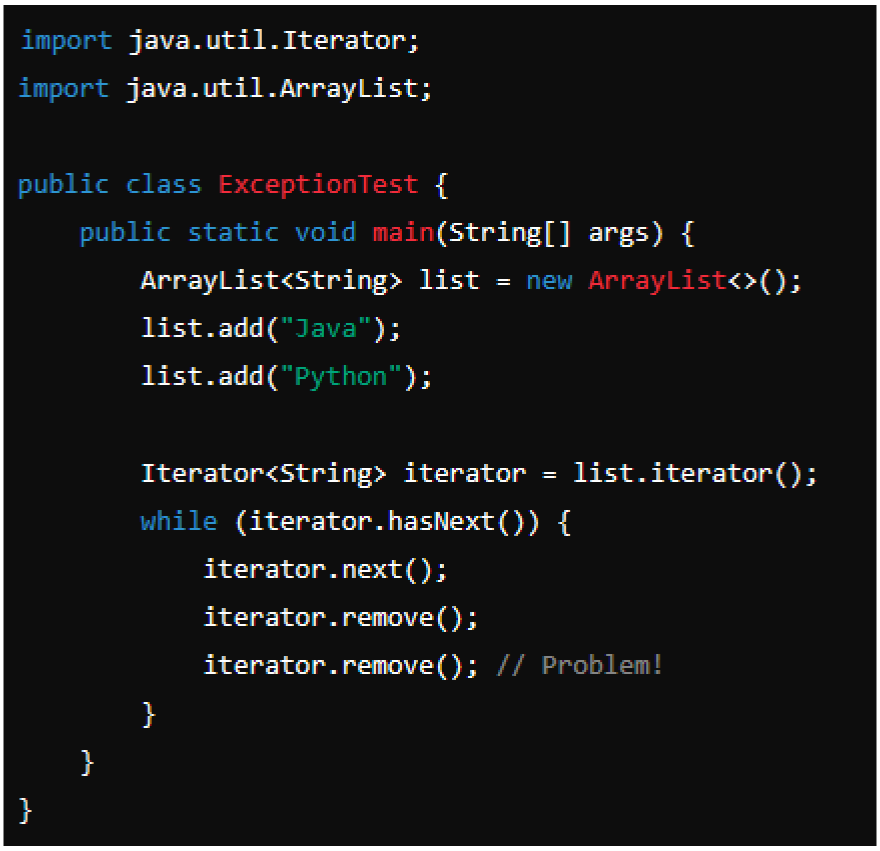
What exception is thrown and why?

1.6

```java
public class ExceptionTest {
    public static void main(String[] args) {
        int result = 10 / 0; // Problem!
        System.out.println(result);
    }
}
```

What exception is thrown and why?

1.7



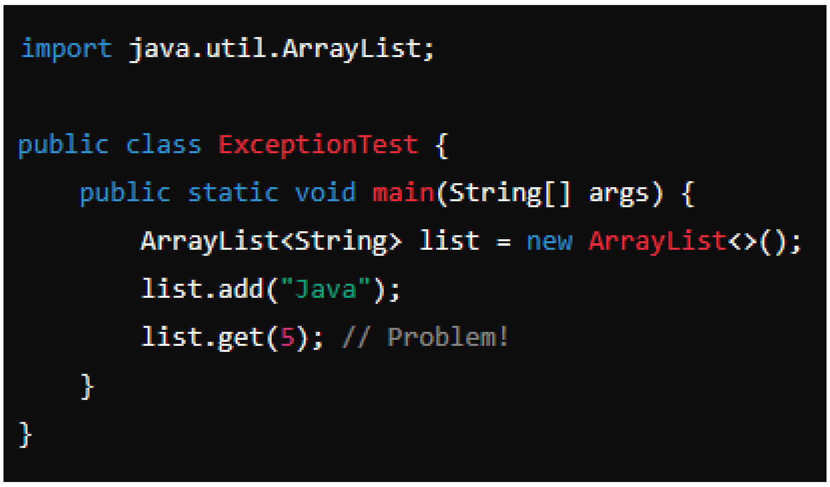
```java
import java.util.Iterator;
import java.util.ArrayList;

public class ExceptionTest {
    public static void main(String[] args) {
        ArrayList<String> list = new ArrayList<>();
        list.add("Java");
        list.add("Python");

        Iterator<String> iterator = list.iterator();
        while (iterator.hasNext()) {
            iterator.next();
            iterator.remove();
            iterator.remove(); // Problem!
        }
    }
}
```

What exception is thrown and why?

1.8

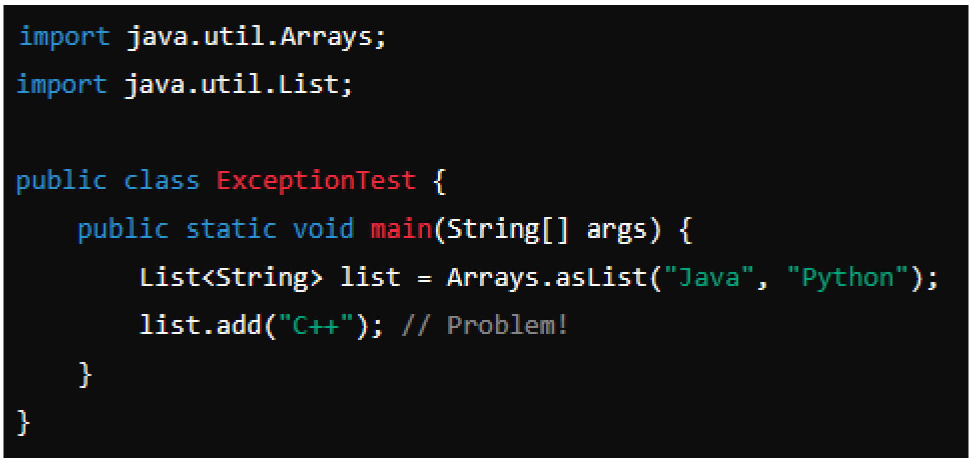


```java
import java.util.ArrayList;

public class ExceptionTest {
    public static void main(String[] args) {
        ArrayList<String> list = new ArrayList<>();
        list.add("Java");
        list.get(5); // Problem!
    }
}
```

What exception is thrown and why?

1.9

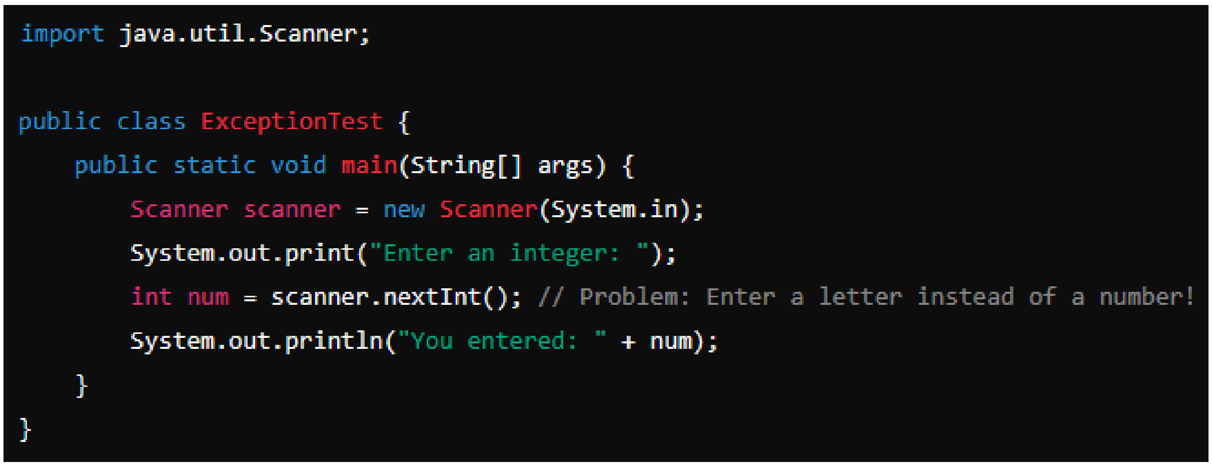


```java
import java.util.Arrays;
import java.util.List;

public class ExceptionTest {
    public static void main(String[] args) {
        List<String> list = Arrays.asList("Java", "Python");
        list.add("C++"); // Problem!
    }
}
```

What exception is thrown and why?

1.10



```java
import java.util.Scanner;

public class ExceptionTest {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter an integer: ");
        int num = scanner.nextInt(); // Problem: Enter a letter instead of a number!
        System.out.println("You entered: " + num);
    }
}
```

What exception is thrown and why?

Aufgabe 2

2.1 What is the difference between strong, weak, and soft references?

2.2 How can you suggest Garbage Collection in Java?

2.3 What is the purpose of Garbage Collection in Java?

2.4 What is the difference between checked and unchecked exceptions?

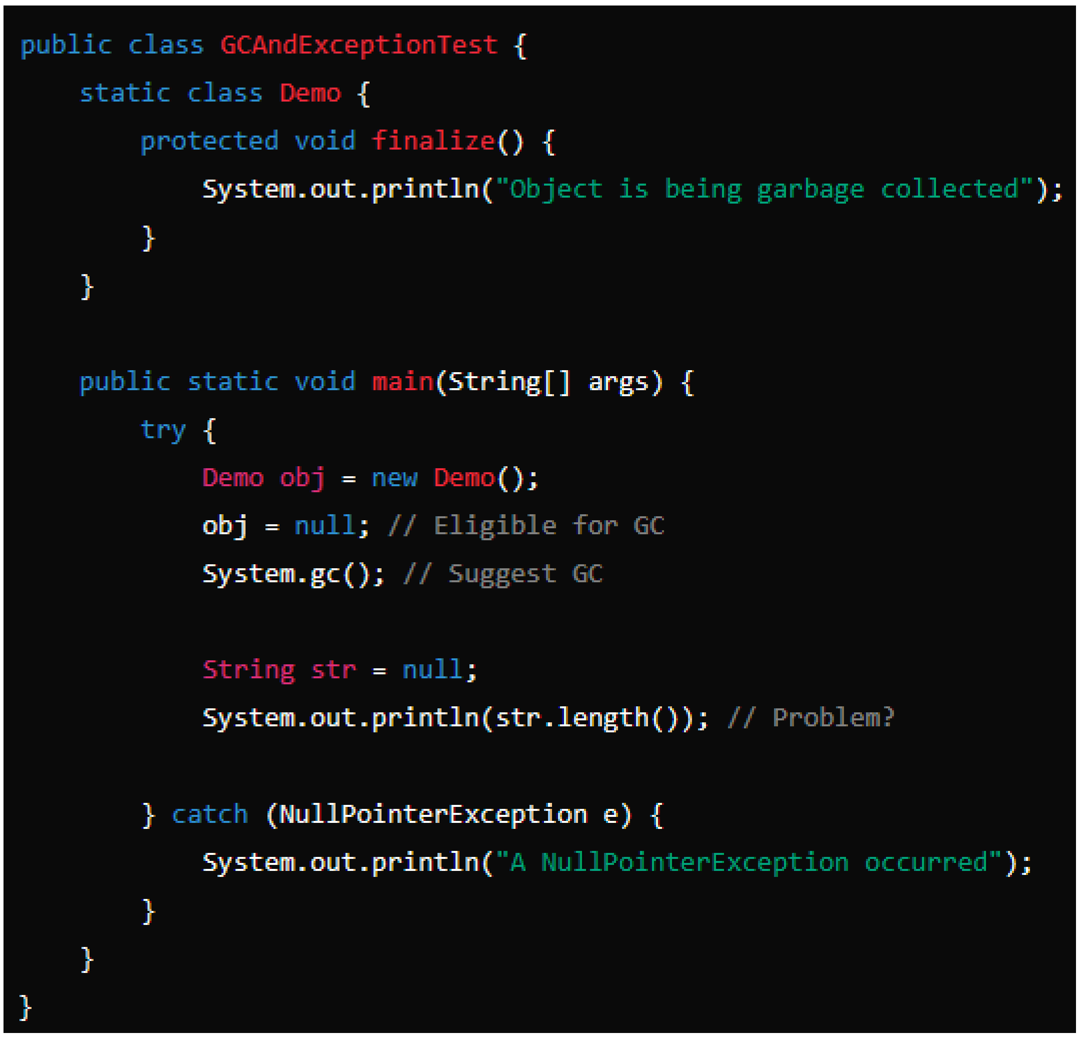
2.5 What happens if an exception is thrown in a `finally` block?

Aufgabe 3

Fill in the Table with checked and unchecked Exceptions

Exception	Type	Short Description
IOException	checked	General I/O failure (e.g., file not found, read error).
	checked	
	checked	
	checked	
	checked	
	unchecked	
	unchecked	
	unchecked	
	unchecked	
	unchecked	

Aufgabe 4



```java
public class GCAndExceptionTest {
    static class Demo {
        protected void finalize() {
            System.out.println("Object is being garbage collected");
        }
    }

    public static void main(String[] args) {
        try {
            Demo obj = new Demo();
            obj = null; // Eligible for GC
            System.gc(); // Suggest GC

            String str = null;
            System.out.println(str.length()); // Problem?

        } catch (NullPointerException e) {
            System.out.println("A NullPointerException occurred");
        }
    }
}
```

4.1 What will be the output?

A) `"Object is being garbage collected"`

B) `"A NullPointerException occurred"`

C) Both A and B (Order may vary)

D) No output

Answer:

```java
public class NumberFormatExample {
    public static void main(String[] args) {
        String validNumber = "123";
        String invalidNumber = "12A3";

        int num1 = Integer.parseInt(validNumber);   // Line 1
        int num2 = Integer.parseInt(invalidNumber); // Line 2
        System.out.println(num1 + num2);
    }
}
```

4.2 Which line in the following code will throw a `NumberFormatException`?

A) Line 1
B) Line 2
C) Both Line 1 and Line 2
D) No Exception

Answer:

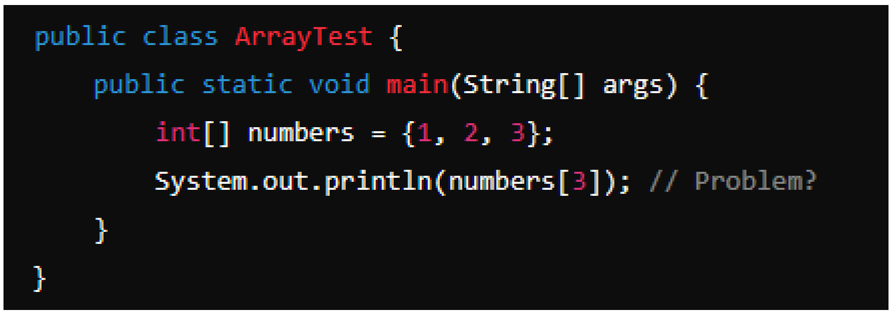
```java
public class Test {
    public static void checkAge(int age) {
        if (age < 0) {
            throw new IllegalArgumentException("Age cannot be negative");
        }
        System.out.println("Valid age: " + age);
    }

    public static void main(String[] args) {
        checkAge(-5); // Problem?
    }
}
```

4.3 What will happen?

A) Prints `"Valid age: -5"`
B) Throws IllegalArgumentException with message `"Age cannot be negative"`
C) Compilation error
D) Prints `"Valid age: 0"`

Answer:

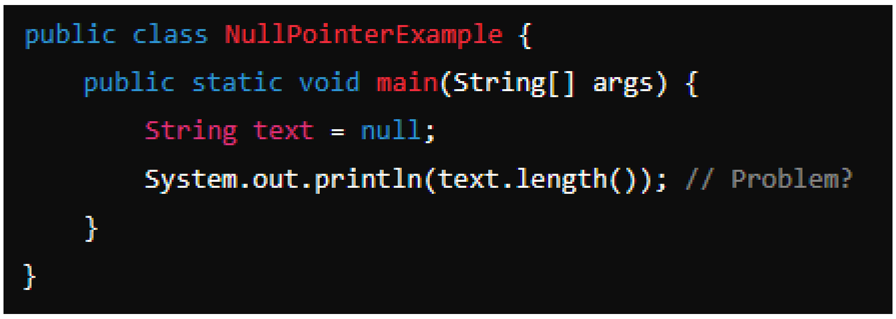


```java
public class ArrayTest {
    public static void main(String[] args) {
        int[] numbers = {1, 2, 3};
        System.out.println(numbers[3]); // Problem?
    }
}
```

4.4 What happens when the following code is executed?

A) Prints 3
B) Prints 0
C) Throws **ArrayIndexOutOfBoundsException**
D) Compilation error

Answer:

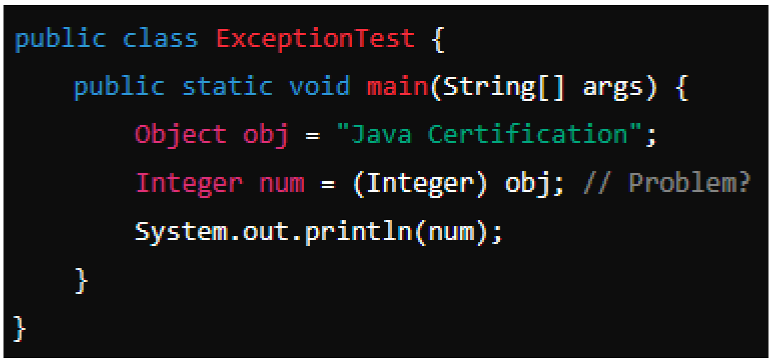


```java
public class NullPointerExample {
    public static void main(String[] args) {
        String text = null;
        System.out.println(text.length()); // Problem?
    }
}
```

4.5 The following program throws a `NullPointerException`. How can you fix it?

A) Replace `null` with `"Hello"`
B) Use `if (text != null)` before accessing `length()`
C) Surround with `try-catch`
D) All of the above

Answer:



```java
public class ExceptionTest {
    public static void main(String[] args) {
        Object obj = "Java Certification";
        Integer num = (Integer) obj; // Problem?
        System.out.println(num);
    }
}
```

4.6 What happens when the following code runs?

A) Prints `"Java Certification"`
B) Prints `null`
C) Throws **ClassCastException** at runtime
D) Compiles successfully and runs without error