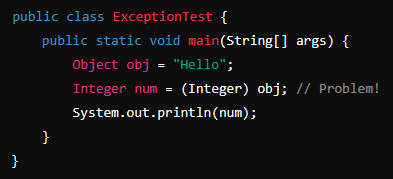
20250211 Pflichtaufgabe

## Aufgabe 1

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

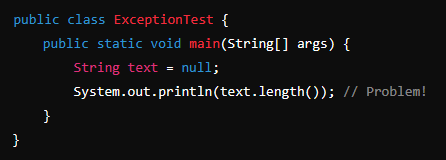
1.1



**What exception is thrown and why?**

I assume it’s going to be a **ClassCastException** (Runtime), because an Object cannot fit into an Integer, although it might contain one, but here it is even a String in it.

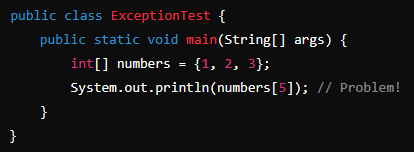
1.2



**What exception is thrown and why?**

There will a NullPointerException because text points at null and there cannot be a length to.

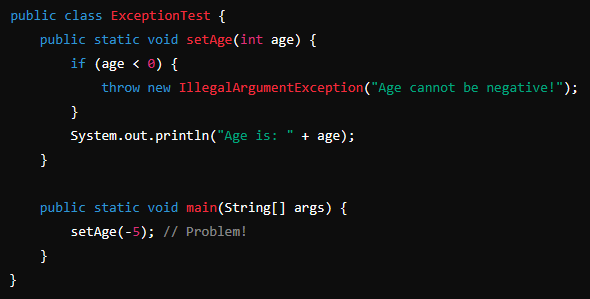
1.3



**What exception is thrown and why?**

ArrayIndexOutOfBoundsException, because the array has the highest index 2 (0-2 / 3)

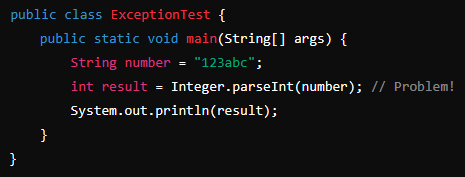
1.4



**What exception is thrown and why?**

IllegalArgumentException as it says in upper method, because the passed age is negative.

1.5

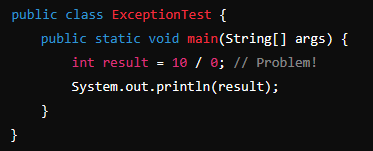


**What exception is thrown and why?**

That might be an NumberFormatException, because abc part can’t be parsed as an integer.

Not an InputMissMatchException, because a String is expected and even mandatory here.

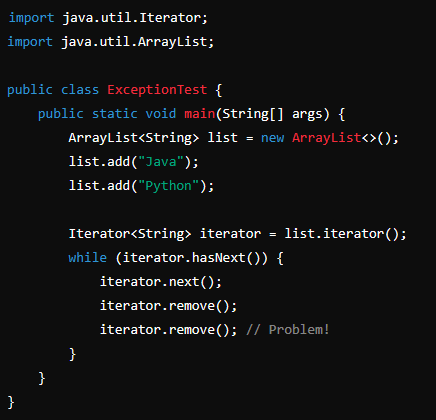
1.6



**What exception is thrown and why?**

ArithmeticException because there is being tried to divide by zero.

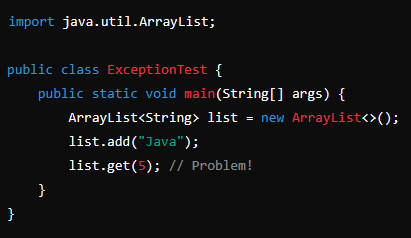
1.7



**What exception is thrown and why?**

ArrayIndexOutOfBoundsException, because first the iterator jumps from entry “Java” to “Python”, removes it and then tries to remove another item, that is not existent.

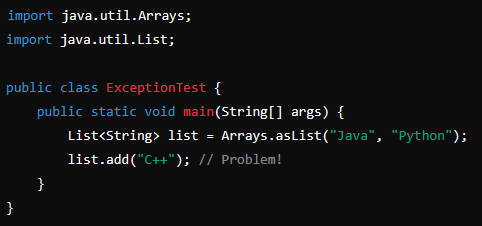
1.8



**What exception is thrown and why?**

ArrayIndexOutOfBoundsException, because there is being tried to access/get the item at the fourth place (from zero), but there is only one item stored in the list. So list.get(0) only works.

1.9



**What exception is thrown and why?**

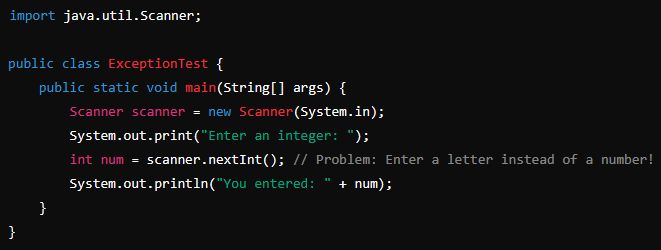
First, I assumed an ArrayIndexOutOfBoundsException, because the list gets passed a fixed sized list on initialization, but I doubted and tried it out.

Turned out it throws a so called **UnsupportedOperationException** (RunTime), looked into the asList-Method and that should return a new ArrayList, that can indeed be fed by add().

But, and here comes the turn, the class referred to as invoking add on the List, follows back to the AbstractList-Class, that implements List-Interface and has an add() throwing this Exception. Only the ArrayList-Class, that inherits from AbstractList, which in turn implements the List-Interface, like its sub, has another add() add-Method not throwing this.

The only thing I wonder about and I would like to get an Explanation for, is why does AbstractList return an ArrayList-Object, that is being assigned to List and add() from AbstractList is used anyway. This is where my confusion starts, and all goes way too far.  
  


1.10



**What exception is thrown and why?**

This will cause an InputMissMatchException, because letters cannot be stored in an integer.

## Aufgabe 2

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

//This was the only thing I didn’t know here. I had to google it.  
<https://medium.com/@davoud.badamchi/explain-strong-weak-soft-and-phantom-references-in-java-their-roles-in-garbage-collection-and-af856ec9257f>

* **Strong References**: Prevent objects from being garbage collected.
* **Weak References**: Allow objects to be collected when there are no strong references, useful for memory-sensitive caches.
* **Soft References**: Allow objects to be collected only when memory is needed, useful for caches where you prefer to retain objects.

My interpretation or what I think I might have understood :D

* strong: Usual, as we have learned, any object that is linked to a reference variable. They are not eligible for garbage collection as long they are not orphaned.
* weak: Can be an object nested inside another, but collected if memory is needed
* soft: like weak but only collected if memory is low and needed

**If this topic is important for the 808, I would like to get informed about it in a lesson.**

**2.2 How can you suggest Garbage Collection in Java?**

By typing System.gc();

**2.3 What is the purpose of Garbage Collection in Java?**

It frees up space from object those are not reachable anymore, to keep the Runtime clean.

**2.4 What is the difference between checked and unchecked exceptions?**

As the names suggest a checked exception must be checked for and be handled. This means, if you intend to throw it anyway ether the method must have it with throws in its signature or it must be handled in a try-catch-block directly. If the method has it in its signature that means a rethrow and that can happen over multiple levels, but at the last method in stack, latest the main, must handle it, if it cannot be thrown further elsewhere.

The unchecked exception however can but needn’t to be thrown, stated in a signature or caught and handled there. It can occur “hovering” and handled, if you feel the need to do so.

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

It will be thrown in any case and even override an Exception that might be thrown before and usually would make the finally-block not reachable, before the program crashes.

## Aufgabe 3

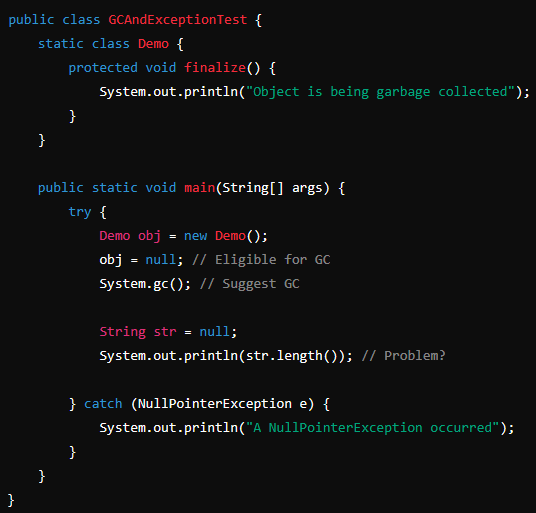
Fill in the Table with checked and unchecked Exceptions

I had some help: <https://www.tpointtech.com/list-of-checked-exceptions-in-java>

|  |  |  |
| --- | --- | --- |
| Exception | Type | Short Description |
| IOException | checked | General I/O failure (e.g., file not found, read error). |
| FileNotFoundException | checked | more specific example. Part of the java.io. package |
| ClassNotFoundException | checked | If there is being tried to refer to a class that cannot be found or doesn’t even exist. |
| SQLException | checked | Occurs if an SQL related action fails. java.sql package |
| ParseException | checked | If a String is parsed as Date/Time, and it fails. java.text content package |
| RunTimeException | unchecked | The mother of all unchecked  and can be used as general |
| ArrayIndexOutOfBounds-Exception | unchecked | If an index not existent is tried to be accessed |
| ArithemitcException | unchecked | Whenever I mathematical error occurs like divide by 0 |
| IllegalArgumentException | unchecked | When a method gets passed wrong or unexpected parameters those couldn’t be avoided by the compiler like a number out of given range |
| InputMissMatchException | unchecked | For example, thrown by scanner, if expected a number but given a String. |

Note: InputMismatchException occurs for any data type mismatch, while NumberFormatException is specific to errors in converting a String to a numeric type.

## Aufgabe 4



**4.1 What will be the output?**

A) "Object is being garbage collected"

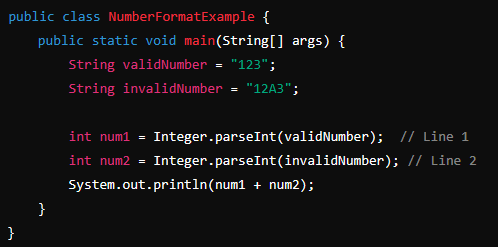
B) "A NullPointerException occurred"

C) Both A and B (Order may vary)

D) No output

Answer:

C) because you cannot say if the finalize() and garbage collection will occur, but they could.  
The NullPointerException will occur in any case, because length of str = null can’t be gotten.



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

A) Line 1

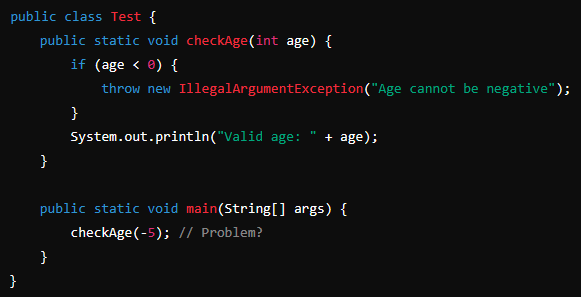
B) Line 2

C) Both Line 1 and Line 2

D) No Exception

Answer:

B) the exception will occur as soon as the Integer-class tries to parse A in 12A3 as integer.



**4.3 What will happen?**

A) Prints "Valid age: -5"

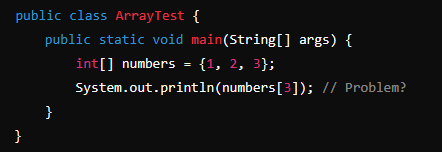
B) Throws IllegalArgumentException with message "Age cannot be negative"

C) Compilation error

D) Prints "Valid age: 0"

Answer:

B) because in checkAge(int age) the passed integer is tested for being smaller then 0 and a parameter of -5 will trigger that if branch and that throws the exception.



**4.4 What happens when the following code is executed?**

A) Prints 3

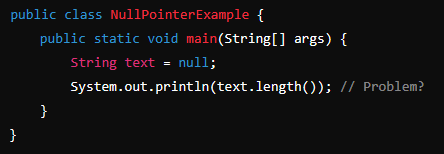
B) Prints 0

C) Throws **ArrayIndexOutOfBoundsException**

D) Compilation error

Answer:

The array numbers has a length of 3, but that also means its highest index number is 2.



**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

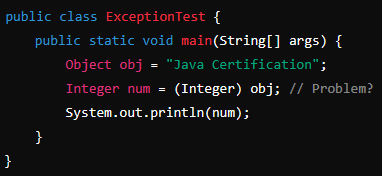
Answer:

D), because

- (A) the String “Hello” has a length, namely 5.

- (B) if text = null, length() won’t be tried to access.

- (C) won’t let crash the program if used properly and NullPointerException will be caught  
So just surrounding with a try-catch won’t guarantee a fix, but can be used for that.



**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

Answer:

As stated all above firstly, a String won’t fit in an Integer by type and even if obj was fed by an integer value, the obj Object is far higher in hierarchy. You cannot simply become your own father or any other ancestor, that makes no sense.