Optional class

Agenda

1. What is NullPointerException?
2. Optional class
3. get( ) method
4. orElseThrow( ) method
5. orElse( ) method
6. isPresent( ) method
7. ifPresent( ) method

What is NullPointerException?

What is NullPointerException?

- When we declare reference variables and do not initialize them, a special value **null** is assigned to them.

Code:

```
public class Employee  {
    String name;
    public static void main(String[ ] args)  {
        Employee obj1 = new Employee( );
        System.out.println(obj1.name);
    }
}
```

Output:

```
null
```

What is NullPointerException?

- When we try to use a reference variable which is initialized to **null** and not holding a valid object, we end up with **NullPointerException** during Runtime.

Code:

```
public class Employee  {
        String name;
        public static void main(String[ ] args)  {
                Employee obj1 = new Employee( );
                char first_char = obj1.name.charAt(0);
                System.out.println(first_char);
        }
}
```

Output:

```
Exception in thread "main"
java.lang.NullPointerException
at Employee.main(Employee.java:5)
```

Optional class

Optional class

- In Java 8, the **Optional** class which provides an easier way to avoid the NullPointerException was introduced.
- Optional class is added in **java.util** package and it is a **final** class.
- It helps in writing neat and more readable code without using too many null checks.
- By using Optional instance, we can specify alternate values to return or alternate code to run.
- **Optional instance:** A container object which may or may not contain a non-null value.

Different ways of using Optional class:

1. Get the value.
2. Get if Object is Not Null, else Throw Exception.
3. Get if Object is Not Null, else return default.
4. Consume if it is not Null.

https://docs.oracle.com/javase/8/docs/api/java/util/Optional.html

get( ) method

get( ) method

Get the value using get( ) method: It simply returns the object from the optional instance. In this case, no null check is done and the optional instance returns the contained value as it is.

Example: 1

```
import java.util.Optional;
public class Employee  {
        String name = "Admin";
        public static void main(String[ ] args)  {
                Employee obj1 = new Employee( );
                Optional<String> n = Optional.ofNullable(obj1.name);
                System.out.print(n.get( ));
        }
}
```

Returns an Optional instance with a present value if the specified value is not null, otherwise an empty Optional instance.

Output:

Admin

get( ) method contd..

Example: 2

```
import java.util.Optional;
public class Employee  {
        String name;
        public static void main(String[ ] args)  {
                Employee obj1 = new Employee( );
                Optional<String> n = Optional.ofNullable(obj1.name);
                System.out.print(n.get( ));
        }
}
```

Output:

```
Exception in thread "main" java.util.NoSuchElementException: No value present at
java.util.Optional.get(Optional.java:135) at Employee.main(Employee.java:7)
```

orElseThrow( ) method

orElseThrow( ) method

Get if Object is Not Null, else Throw Exception using orElseThrow( ) method: It returns the object value if the object is not null and if the object is null it throws the specified exception.

Example: 1

```
import java.util.Optional;
public class Employee  {
        String name = "Admin";
        public static void main(String[ ] args)  {
                Employee obj1 = new Employee( );
                Optional<String> n = Optional.ofNullable(obj1.name);
                System.out.print( n.orElseThrow( NullPointerException::new ) );
        }
}
```

Output:

```
Admin
```

orElseThrow( ) method contd..

Example: 2

```
import java.util.Optional;
public class Employee  {
        String name;
        public static void main(String[ ] args)  {
                Employee obj1 = new Employee( );
                Optional<String> n = Optional.ofNullable(obj1.name);
                System.out.print(n);
        }
}
```

Output:

```
        Optional.empty
```

orElseThrow( ) method contd..

Example: 3

```
import java.util.Optional;
public class Employee  {
        String name;
        public static void main(String[ ] args)  {
                Employee obj1 = new Employee( );
                Optional<String> n = Optional.ofNullable(obj1.name);
                System.out.print( n.orElseThrow( NullPointerException::new ) );
        }
}
```

Output:

Exception in thread "main" java.lang.NullPointerException at java.util.Optional.orElseThrow(Optional.java:290) at Employee.main(Employee.java:7)

orElse( ) method

orElse( ) method

Get if Object is Not Null, else return default value using orElse( ) method: It returns the contained value if it is not null. Otherwise it returns the given default value.

Example: 1

```
import java.util.Optional;
public class Employee  {
        String name = "Admin";
        String default_name = "User";
        public static void main(String[ ] args)  {
                Employee obj1 = new Employee( );
                Optional<String> n = Optional.ofNullable(obj1.name);
                System.out.print( n.orElse( obj1.default_name ) );
        }
}
```

Output:

```
Admin
```

orElse( ) method contd..

Example: 2

```
import java.util.Optional;
public class Employee  {
        String name;
        String default_name = "User";
        public static void main(String[ ] args)  {
                Employee obj1 = new Employee( );
                Optional<String> n = Optional.ofNullable(obj1.name);
                System.out.print( n.orElse( obj1.default_name ) );
        }
}
```

Output:

User

isPresent( ) method

isPresent( ) method

Check if it is Not Null using isPresent( ) method: It returns true if the contained object is not null.

Example: 1

```
import java.util.Optional;
public class Employee  {
        String name = "Admin";
        public static void main(String[ ] args)  {
                Employee obj1 = new Employee( );
                Optional<String> n = Optional.ofNullable(obj1.name);
                if( n.isPresent( ) )  {
                        System.out.print( n.get( ) );
                }  else {
                        System.out.print("Null value");
                }
} }
```

Output:

Admin

isPresent( ) method contd..

Example: 2

```
import java.util.Optional;
public class Employee  {
        String name;
        public static void main(String[ ] args)  {
                Employee obj1 = new Employee( );
                Optional<String> n = Optional.ofNullable(obj1.name);
                if( n.isPresent( ) )  {
                        System.out.print( n.get( ) );
                }  else {
                        System.out.print("Null value");
                }
        }
}
```

Output:

Null value

ifPresent( ) method

ifPresent( ) method

Consume if it is not Null using ifPresent( ) method: If a value is present, invoke the specified consumer with the value, otherwise do nothing.

Example: 1

```
import java.util.Optional;
public class Employee  {
        String name = "Admin";
        public static void main(String[ ] args)  {
                Employee obj1 = new Employee( );
                Optional<String> n = Optional.ofNullable(obj1.name);
                n.ifPresent( System.out::println );
        }
}
```

Output:

```
Admin
```

ifPresent( ) method contd..

- This code doesn’t print anything.

Example: 2

```
import java.util.Optional;
public class Employee  {
        String name;
        public static void main(String[ ] args)  {
                Employee obj1 = new Employee( );
                Optional<String> n = Optional.ofNullable(obj1.name);
                n.ifPresent( System.out::println );
        }
}
```

Thank you