Java I

Урок VII

Java Exception Handling

Что такое exception?

Что такое exception?

- Exception это нежелательное событие, которое прерывает обычный поток программы
- Плохие новости:
  - Что-то пошло не так, и возможно программа умрет
- Хорошие новости:
  - Мы можем это предотвратить

Пример Exception

```
String a = null;
System.out.println(a.equals(""));
```

Output:

```
Exception in thread "main" java.lang.NullPointerException
```

Примеры exception классов

- NullPointerException
- ArrayIndexOutOfBoundsException
- FileNotFoundException

try-catch

Как поймать Exception? try-catch блок

```
try {
    // код для выполнения
} catch (exception_type arg) {
    // код при ошибке типа exception_type в блоке try
} catch (other_exception_type arg) {
    // код при ошибке типа other_exception_type в блоке try
}
```

Пример try-catch

```
String a = null;

try {
  System.out.println(a.equals(""));
} catch (NullPointerException npe) {
  System.out.println("a is not defined. Try later...");
}
```

Output:

```
a is not defined. Try later...
```

Как поймать Exception? try-catch-finally блок

```
try {
    // код для выполнения
} catch (exception_type arg) {
    // код при ошибке типа exception_type в блоке try
} catch (other_exception_type arg) {
    // код при ошибке типа other_exception_type в блоке try
} finally {
    // код которые выполнится всегда, не важно, случилась ошибка или нет
}
```

Пример try-catch-finally

```
String a = null;
try {
  System.out.println(a.equals(""));
} catch (NullPointerException npe) {
  System.out.println("a is not defined. Try later...");
} finally {
  System.out.println("cleaning resources");
}
```

Output:

```
a is not defined. Try later...
cleaning resources
```

Checked/unchecked exceptions

Unchecked exceptions

- Unchecked exceptions - компилятор не проверяет код на наличие перехвата ошибок.
- Возникают при некорректном программном коде или невалидных параметрах, поданных в метод.
- Мы не обязаны перехватывать эти ошибки, так как они свидетельствуют о том, что сам код должен быть подправлен.
- Все Unchecked exceptions наследуют класс **RuntimeException**

Пример unchecked exception

```
String a = null;

try {
  System.out.println(a.equals(""));
} catch (NullPointerException npe) {
  System.out.println("a is not defined. Try later...");
}
```

- try-catch здесь необязателен, код будет компилироваться и без него. Но если мы предполагаем, что здесь может произойти какая-то ошибка, мы добавляем try-catch

Checked exceptions

- Checked exceptions - если какой-либо метод кидает checked exceptions, то компилятор проверяет, чтоб программист не забыл **перехватить** эту ошибку или **прокинуть** выше.
- Возникают из-за каких-то факторов вне нашего приложения (файл не был найден, сервер недоступен и т.д.)

Checked exceptions

```
private void readFileWithTry() {
  File file = new File("test.txt");
  FileReader reader = new FileReader(file);
}
```

- Этот код не будет компилироваться, так как создание нового FileReader может выкинуть checked exception - FileNotFoundException.
- Мы обязаны либо добавить try-catch, либо объявить, что этот метод тоже кидает checked exception - FileNotFoundException.

Checked exceptions - try-catch

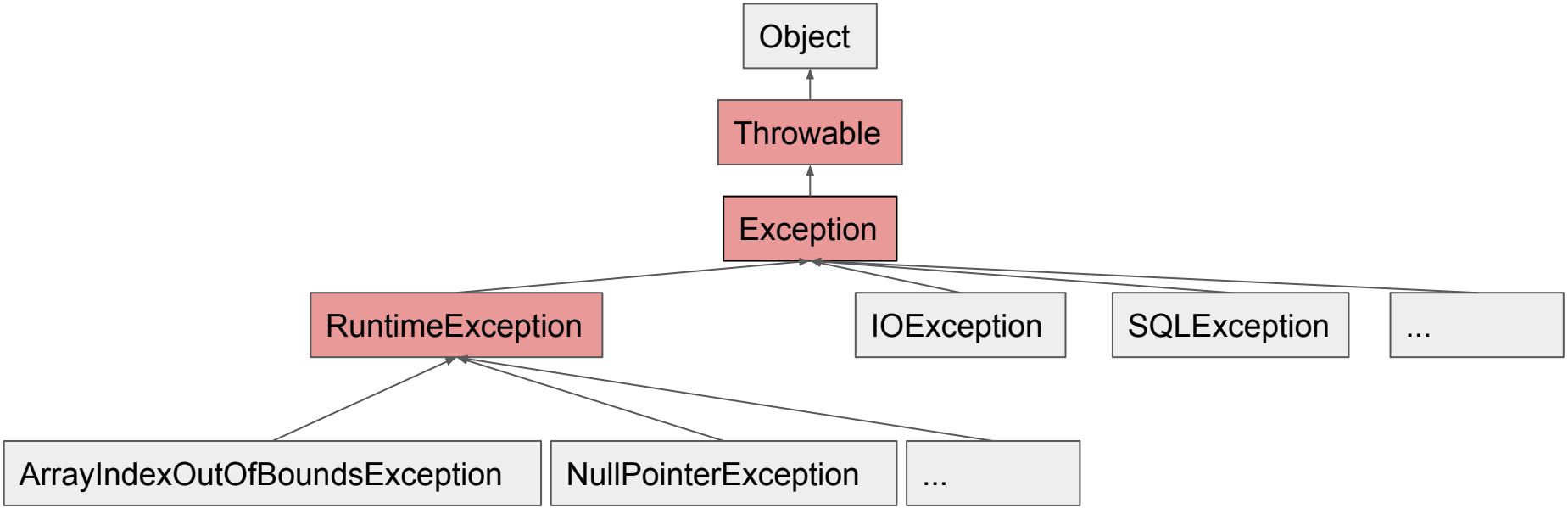
```
private void readFileWithTry() {
  File file = new File("test.txt");
  try {
     FileReader reader = new FileReader(file);
  } catch (FileNotFoundException e) {
     e.printStackTrace();
  }
}
```

Checked exceptions - throws

```
private void readFileWithThrows() throws FileNotFoundException {
  File file = new File("test.txt");
  FileReader reader = new FileReader(file);
}
```

Иерархия exception классов

Иерархия exception классов



Перехват всех ошибок

```
try {
  doSmth(); //any exception may occur
} catch (Throwable e) {
  e.printStackTrace();
}
```

Throwing Exceptions

Как кинуть ошибку?

```
throw new Exception(“Message”)
```

Throwing checked exception

```
public static void main(String[] args) throws Exception {
  Scanner scanner = new Scanner(System.in);
  int input = scanner.nextInt();

  if (input < 0) {
     throw new Exception("You have entered invalid value");
  }
  System.out.println(input);
}
```

Throwing unchecked exception

```
public static void main(String[] args) {
  Scanner scanner = new Scanner(System.in);
  int input = scanner.nextInt();

  if (input < 0) {
     throw new RuntimeException("You have entered invalid value");
  }
  System.out.println(input);
}
```

Создание собственных Exception-ов

Создание checked exception

```
public class CustomCheckedException extends Exception {

  public CustomCheckedException(String message) {
     super(message);
  }

}
```

```
public void doSmth() throws CustomCheckedException {
    ...
    throw new CustomCheckedException("Test");
    ...
}
```

Создание unchecked exception

```
public class CustomUncheckedException extends RuntimeException {

  public CustomUncheckedException(String message) {
     super(message);
  }

}
```

```
public void doSmth() {
    ...
    throw new CustomUncheckedException("Test");
    ...
}
```

String

Создание String

```
String str1 = "Welcome";
String str2 = "Welcome";
```

- Если объект уже есть в памяти, то он не создает новый, а присваивает его же к другой переменной. В итоге в памяти образуется один объект “Welcome” и две переменных, ссылающихся на него.

Создание String

```
String str1 = new String("Welcome");
String str2 = new String("Welcome");
```

- Используя new String() в памяти каждый раз создается новый объект

Создание и сравнение String

```
String str1 = "Welcome";
String str2 = "Welcome";
System.out.println(str1 == str2);          //true
System.out.println(str1.equals(str2));     //true

String str3 = new String("Welcome");
String str4 = new String("Welcome");
System.out.println(str3 == str4);          //false
System.out.println(str3.equals(str4));     //true
```

Методы String

https://docs.oracle.com/javase/8/docs/api/java/lang/String.html

Сложение строк

concat(String other)

```
String attention = "Hello, ";
String s = attention + "world";
```

OR

```
String s = attention.concat("world");
```

Узнать длину строки

length()

```
String s = "Write once, run anywhere.";
int len = s.length();
```

Выбрать символы из строки

charAt(int ind)

```
String s = "Write once, run anywhere.";
char ch = s.charAt(3);
```

Выбрать подстроку

substring(int begin)

substring(int begin, int end)

```
String s = "Write once, run anywhere.";
String sub1 = s.substring(6, 10); //once
String sub2 = s.substring(16);    //anywhere.
```

Разбить строку на подстроки

split(String regExp)

```
String s = "Write:once,:run:anywhere.";
String[] sub = s.split(":");
```

Сравнить строки

equals(Object obj)

equalsIgnoreCase(Object obj)

```
String s2 = "Другая строка";
s2.equals("другая строка");                  //false
s2.equalsIgnoreCase("другая строка");   //true
```

Найти подстроку

indexOf(String sub)

indexOf(int ch)

```
"Молоко".indexOf('o');    //1
"Молоко".indexOf('a');    //-1
"Молоко".indexOf("лок");  //2
```

Изменить регистр букв

`toLowerCase()`

`toUpperCase()`

Заменить подстроку

```
replace(String old, String new)
```

Убрать пробелы в начале и конце строки

`trim()`

Преобразовать в строку данные другого типа

```
valueOf(type elem)
```