

Java

Основы языка. Часть 2

Сергей Рыбалкин

Отметьтесь на портале.

- + Web services
 - + Databases access application
 - + Highload systems
 - + Low latency systems
 - + Distributed systems
 - + JVM
- CPU bound applications
 - Hardware-access systems



Система контроля версий. Git



Классы. Инстанцирование



Исключения



Generics



Система контроля версий. Git



Классы. Инстанцирование



Исключения



Generics

<https://git-scm.com/downloads>

Нужно добавить git в Path

git clone <url>

Создает локальную копию удаленного репозитория

Например,

```
git clone https://github.com/rybalkinsd/atom.git
```

git checkout <branch>

Переключается на ветку.

Например,

```
git checkout master
```


`git checkout -b <name>`

Создает новую ветку и переключается на нее.

Например,

```
git checkout -b feature
```

git add <file1 file2 ...>

Помечает файлы для сохранения.

Например,

```
git add .
```

```
git add -all
```

```
git add Main.java
```

`git commit -m "Message"`

Сохраняет помеченные файлы.

Например,

```
git commit -m "bug #312 NPE fix"
```

git push <to> <branch>

Отправляет изменения из локального репозитория в удаленный.

Например,

```
git push origin master
```

```
git push
```

git pull

Затягивает изменения из удаленной ветки в локальную

git status

Отображает изменения в локальном репозитории

Полная документация:

<https://git-scm.com>

Почитать:

<https://habrahabr.ru/post/141160/>



Система контроля версий. Git



Классы. Инстанцирование



Исключения



Generics

Override — позволяет классу наследнику предоставлять реализацию метода, ранее описанного или задекларированного в базовом классе или интерфейсе.

@Override

Overload — позволяет классу иметь несколько методов с одинаковым именем и разным набором параметров.

Override — выбор исполняемого метода осуществляется на этапе исполнения

Overload — выбор осуществляется в момент компиляции.

```
enum Gender {  
    MALE,  
    FEMALE,  
    OTHER  
}
```

Нет наследования.

Могут реализовывать интерфейсы.

```
T val = new T();
```

```
T val = new T(arg1, arg2, ...);
```

Например,

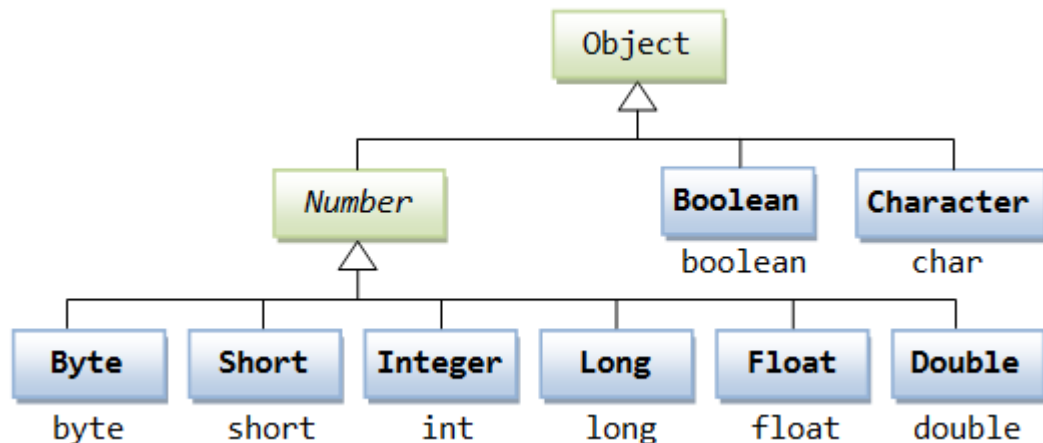
```
File file = new File("input.txt");
```

```
class sub extends Base { }  
Base instance = new Sub();
```

```
class ApiImpl implements Api { }  
Api instance = new ApiImpl();
```

@see `ru.atom.samples.instantiation`

Классы. Primitive types



- immutable types
- boxing / unboxing



Система контроля версий. Git



Классы. Инстанцирование



Исключения

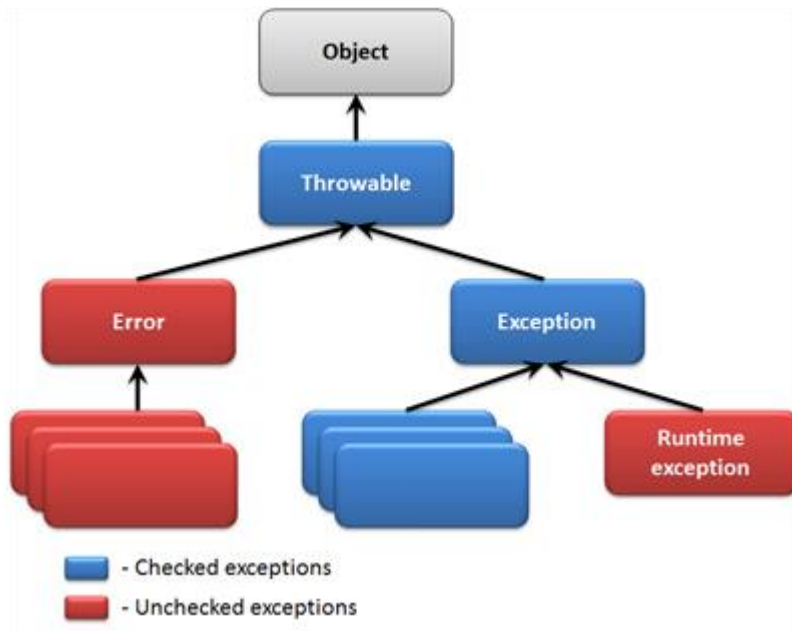


Generics

Всегда что-то может пойти не так.

Нет соединения/ некорректное состояние /
кончилась память/ файл не найден/ ...

Некоторые проблемы в системе можно починить.



- `Throwable(
String message,
Throwable cause)`
- `getStackTrace()`
- `getMessage()`
- `getCause()`

```
try {  
    statements  
} catch (ExceptionType e) {  
    statements  
} finally {  
    statements  
}
```

@see `ru.atom.samples.exception`



Система контроля версий. Git



Классы. Инстанцирование



Исключения



Generics

- Позволяют реализовывать обобщенные алгоритмы
- Обеспечивают безопасное приведение типов
- Убирают из кода явные приведения
- Обрабатываются в compile time. В runtime отсутствуют*


```
class Box<T> {  
    private T t;  
    public void set(T t) {  
        this.t = t;  
    }  
    public T get() {  
        return t;  
    }  
}
```

```
classClazz {  
    public <T> void foo(T t) {  
        ...  
    }  
}
```

Wildcards

Type bounds

Type erasure

Bridge method

Raw-type

`<T extends A & B & C>`

`<? extends A>` - wildcard с ограничениями

`<? super A>` - wildcard с ограничениями

`<?>` - wildcard без ограничений

`Class<Object>` не является предком `Class<Integer>`

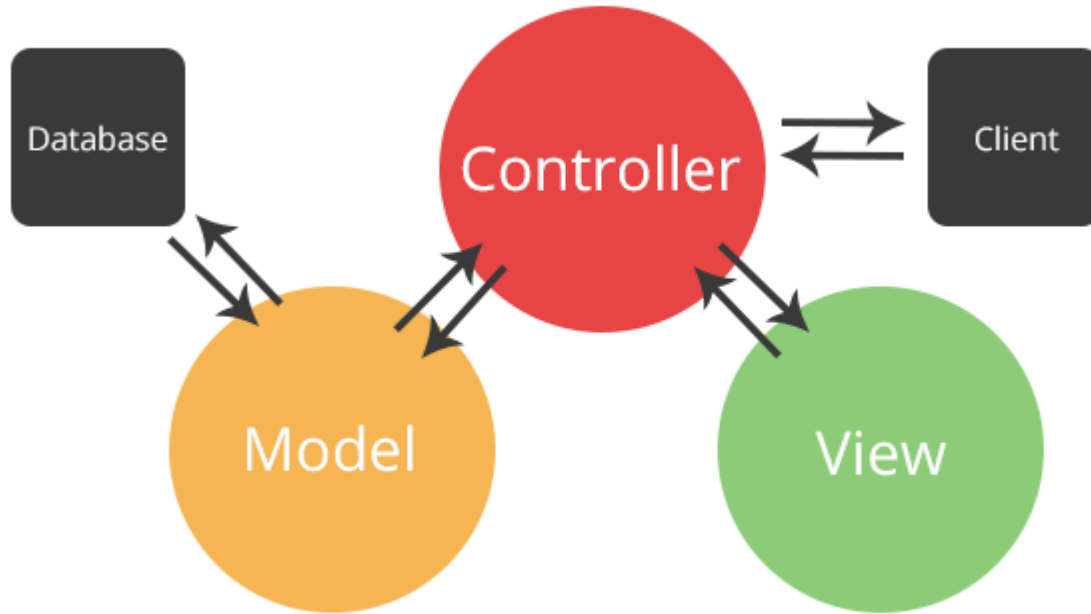
@see `ru.atom.samples.generics`

Оставьте отзыв.

Спасибо за внимание!

Сергей Рыбалкин

s.rybalkin@corp.mail.ru




```
{
  "firstName": "John",
  "lastName": "Smith",
  "isAlive": true,
  "age": 25,
  "address": {
    "streetAddress": "21 2nd Street",
    "city": "New York",
    "state": "NY",
    "postalCode": "10021-3100"
  },
  "phoneNumbers": [
    {
      "type": "home",
      "number": "212 555-1234"
    },
    {
      "type": "office",
      "number": "646 555-4567"
    },
    {
      "type": "mobile",
      "number": "123 456-7890"
    }
  ],
  "children": [],
  "spouse": null
}
```

```
<!DOCTYPE html>
<html>
<body>
<h1 style="color: #5e9ca0;">Tinder</h1>
<h2 style="color: #2e6c80;">Profile</h2>
  <p></p>
  <p>Name</p>
  <p><strong>Age: </strong>59</p>
  <p><strong>Description:</strong> Actor and writer currently working on new TV comedy.</p>
  <p><a href="https://www.instagram.com/">Instagram</a></p>
  <p><span style="background-color: #FF0000; color: #fff;
    display: inline-block; padding: 3px 10px; font-weight: bold; border-radius: 5px;">Nope</span>
    <span style="background-color: #009933; color: #fff; display: inline-block; padding: 3px 10px;
    font-weight: bold; border-radius: 5px;">Like</span>
  </p>
</body>
</html>
```