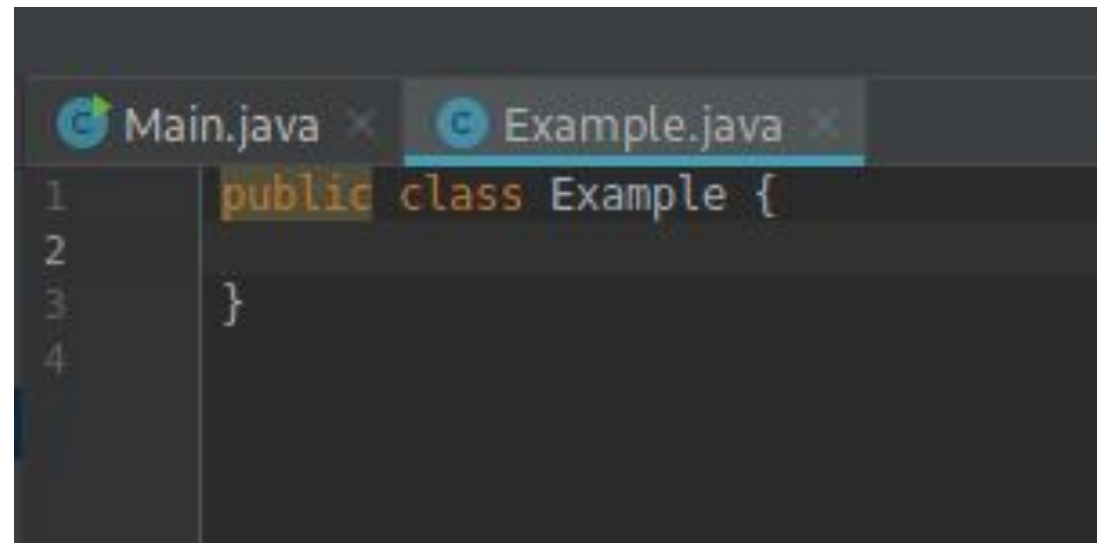


# ООП Java

Продолжение



```
1 public class Example {  
2  
3 }  
4
```

```
Main.java Example.java x
1 public class Main {
2     public static void main(String[] args) {
3         Example example = new Example();
4         example.
5     }
6 }
7
```

m	equals(Object obj)	boolean
m	hashCode()	int
m	toString()	String
m	getClass()	Class<? extends Example>
m	notify()	void
m	notifyAll()	void
m	wait()	void
m	wait(long timeoutMillis)	void
m	wait(long timeoutMillis, int nanos)	void
	arg	functionCall(expr)
	cast	((SomeType) expr)
	capture	T expr ~ (T)expr

Ctrl+Down and Ctrl+Up will move caret down and up in the editor. >>

Почему так?

# Класс Object

- `toString` - превращает объект в строку
- `equals` - сравнивает объекты
- `hashCode` - возвращает хэш объекта
- `getClass` - возвращает класс объекта (об этом попозже)

# toString

```
public class User {  
    private String username;  
    private String password;  
  
    public User(String username, String password) {  
        this.username = username;  
        this.password = password;  
    }  
  
    @Override  
    public String toString() {  
        return "User{" +  
            "username='" + username + '\'' +  
            "}'";  
    }  
}
```

# Как и с чем есть equals?

- Сравнение объектов
- Если `a.equals(b)`, то `b.equals(a)`
- `a.equals(a)` true
- Если `a.equals(b)`, `b.equals(c)`, то `a.equals(c)`
- `a.equals(null)` - false
- Если объекты не изменились, `equals` возвращает один и тот же результат

# hashCode

- Выводит хеш-код — это число int
- Для равных объектов хеш всегда равный
- Для равных объектов, хеш разный, за исключением коллизий
- Простая хеш-функция — сложить все буквы в строке и взять остаток от деления на 10

Пример

abc=

abd=

cab=



# Практика

- Создать иерархию из 1 предка и 2х наследников предка (бытовая техника, музыкальные инструменты)
- Переопределить equals и hashCode в одном наследнике
- Во всех наследниках переопределить toString
- У предка должен быть 1 публичный метод
- У одного из наследников создать приватный метод. Использовать его внутри публичного