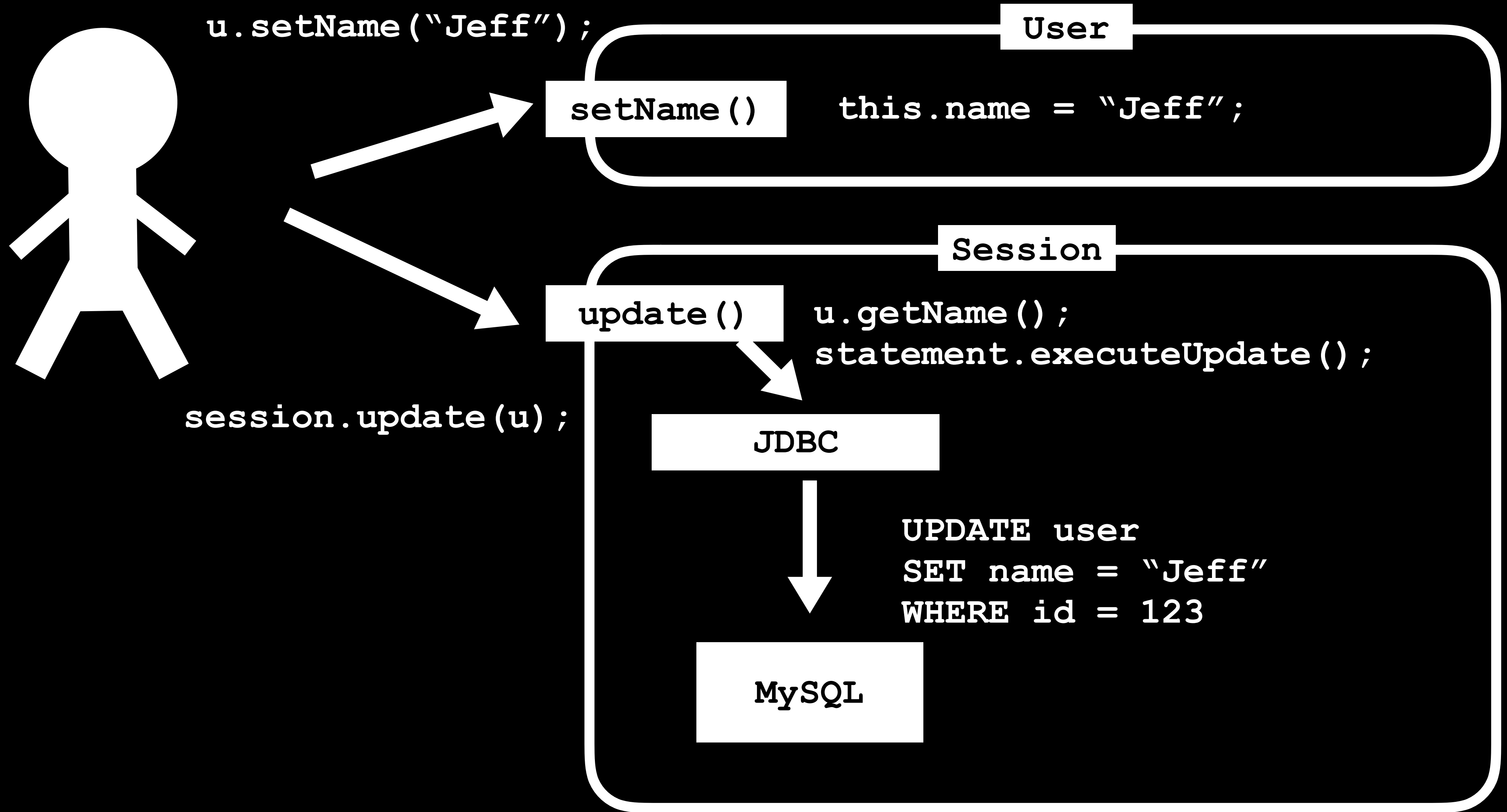


ORM — ЭТО ОБИДНО

ORM is an offensive anti-pattern

Егор Бугаенко

```
@Entity
@Table(name = "post")
public class Post {
    private int id;
    private String title;
    @Id
    @GeneratedValue
    public int getId() {
        return this.id;
    }
    @Column(name = "title")
    public Title getTitle() {
        return this.title;
    }
    public void setTitle(String txt) {
        this.title = txt;
    }
}
```



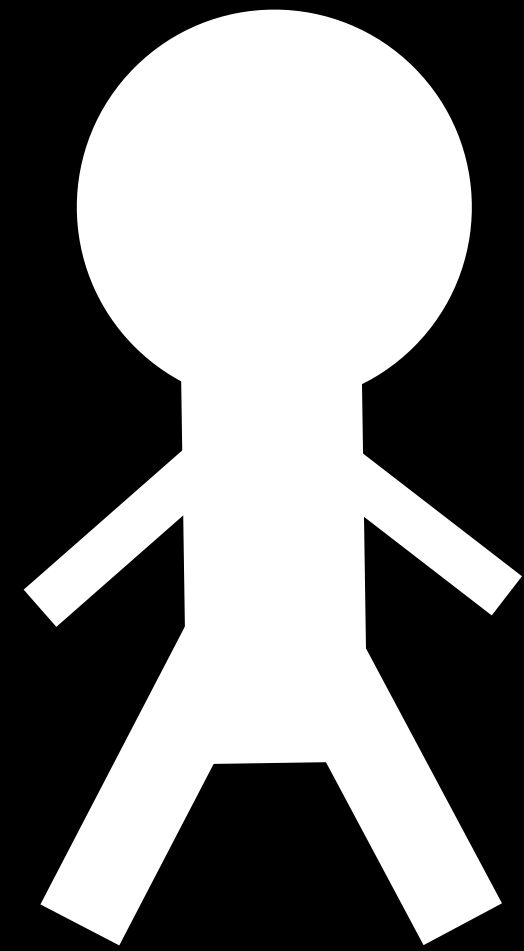
не об'єкт

```
Session session = factory.openSession();
try {
    Transaction txn = session.beginTransaction();
    User user = query.list().get(0);
    user.setName(/* the logic to test */);
    session.update(user);
    txn.commit();
} catch (HibernateException ex) {
    txn.rollback();
} finally {
    session.close();
}
```

```
public class LoggedUser implements User {
    private final User origin;
    public LoggedUser(User user) {
        this.origin = user;
    }
    @Override
    public void rename(String name) {
        long start = System.currentTimeMillis();
        this.origin.rename(name);
        logger.debug(
            "user renaming took %d ms",
            System.currentTimeMillis() - start
        );
    }
}
```

```
List posts = session.createQuery(
    "FROM Post WHERE author = 'Jeff'"
).list();
for (Post post : (List<Post>) posts){
    System.out.println(
        "Title: " + post.getTitle()
    );
}
```

SQL-speaking объекты



```
u.rename("Jeff");
```

```
rename()
```

```
jooq
```

```
statement.executeUpdate();
```

```
JDBC
```

```
UPDATE user  
SET name = "Jeff"  
WHERE id = 123
```

```
MySQL
```

```
User
```

```
x.update("user")  
  .set("name", "Jeff")  
  .where("id", 123)  
  .execute();
```

```
public class Post {
    private int id;
    public int getId() {
        return this.id;
    }
    public Title getTitle() {
        return this.x.select("title")
            .from("post")
            .where("id", this.id)
            .fetchOne();
    }
    public void setTitle(String txt) {
        this.x.update("user")
            .set("title", txt)
            .where("id", this.id)
            .execute();
    }
}
```



Elegant Objects

by Yegor Bugayenko

TL;DR There are 23 practical recommendations for object-oriented programmers. Most of them are completely against everything you've read in other books. For example, static methods, NULL references, getters, setters, and mutable classes are called evil.

0.1
December 22, 2015

www.yegor256.com