

Список використаних джерел

1. C# tutorial. URL: <https://www.w3schools.com/cs/index.php>
2. Introduction to C#. URL: <https://learn.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/tutorials/>
3. Learn C# Tutorial - javatpoint. URL: <https://www.javatpoint.com/c-sharp-tutorial>
4. Антонов Ю.С. Особливості використання елементів системного програмування при роботі з ір-адресами та масками підмережі. Збірники наукових праць професорсько-викладацького складу ДонНУ імені Василя Стуса. Матеріали наукової конференції (2019–2020 рр.) URL: <https://jpvs.donnu.edu.ua/article/view/10405>
5. Антонов Ю.С. Автоматизації перевірки програмного поду на C# студентами та викладачами під час вивчення дисциплін Програмування/Технології програмування. Collection of Scientific Papers «SCIENTIA», (October 7, 2022; Chicago, USA), pp 53–55. URL: <https://previous.scientia.report/index.php/archive/article/view/478>
6. What is C#. URL: <https://www.c-sharpcorner.com/article/what-is-c-sharp/>
7. What is C# Used for?. URL: www.educative.io/answers/what-is-c-used-for

УДК 004.04

Горяшин А.С асистент
кафедри інформаційних технологій

АНАЛІЗ ПРИЧИН ПОШИРЕНOSTІ ORM-ТЕХНОЛОГІЙ У ПРОГРАМНИХ ДОДАТКАХ МОВОЮ JAVA

Донецький національний університет імені Василя Стуса, м. Вінниця

Сьогодні, технології ORM, та різноманітні фреймворки, що реалізують цю специфікацію набувають в різних проектах усе більшої і більшої популярності. Для кожної мови програмування є свої фреймворки для реалізації цієї специфікації. Для мови Java АПІ, що описує правила реалізації ORM – технологій, має назву JPA, а серед певної кількості бібліотек та фреймворків, що реалізують JPA найбільшого поширення набув Hibernate [1].

Взагалі, тема JPA та Hibernate технологій варта окремої книги (можливо і не одної), в цій роботі буде наведено переваги та недоліки використання Hibernate на проектах, та зроблено аналіз щодо рівня необхідності підвищення рівня використання даних технологій у програмних продуктах, що працюють з базами даних.

В ORM-підході є власні переваги та недоліки. Серед переваг можна назвати набагато зручнішу взаємодію між БД та програмним кодом, ніж за допомогою стандартного JDBC (на прикладі Java).

Нехай у нас є деяка таблиця post, що має наступний вигляд:

id	date	title
9	10/24/2014	How to cook a sandwich
13	11/03/2014	My favorite movies
27	11/17/2014	How much I love my job

Рисунок 1 – Приклад таблиці БД

В програмному коді описано клас для цієї таблиці. Він має наступний вигляд:

```
@Entity
@Table(name = "post")
public class Post {
    private int id;
    private Date date;
    private String title;
    @Id
    @GeneratedValue
    public int getId() {
        return this.id;
    }
    @Temporal(TemporalType.TIMESTAMP)
    public Date getDate() {
        return this.date;
    }
    public String getTitle() {
        return this.title;
    }
    public void setDate(Date when) {
        this.date = when;
    }
    public void setTitle(String txt) {
        this.title = txt;
    }
}
```

Рисунок 2 – Приклад класу з анотаціями

В Hibernate головними сутностями є клас та таблиця з бази даних. Вони зв'язуються між собою за допомогою анотацій. Найбільша перевага в тому, що у випадках, коли програміст хоче, наприклад, отримати усі записи з вищенаведеної таблиці, він пише наступний код:

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

Рисунок 3 – Приклад взаємодії через можливості Hibernate

Як можемо бачити, нам не потрібна величезна кількість коду, як в JDBC, щоб прописувати такі класи, як Statement, ResultSet і тд. Менша кількість коду робить процес розробки швидшим, що позитивно впливає на дохід від проекту.

Серед інших переваг можна навести:

- Автоматичне створення SQL-запитів;
- ORM дозволяє ізолювати код програми від деталей зберігання даних;
- Відсутність необхідності створення додаткових SQL-запитів при переносі системи на іншу СУБД;
- Підтримка відображення наслідування і композиції на таблиці;

Hibernate, та інші ORM-фреймворки вже тривалий час зустрічаються майже на всіх java-проектах та у вакансіях програмістів. Без знання даного фреймворку майже нереально влаштуватися на роботу. Але чи настільки він ідеальний?

Очевидно, що у даної специфікації існує ряд недоліків, які необхідно оглянути [3]. Серед них:

- ORM працює повільніше, ніж JDBC;
- Оновлення БД дещо ускладнене;
- Код з використанням Hibernate важко тестувати.

Отже, підсумовуючи сказане, можна зробити висновок, що знання Hibernate сильно впливають на рівень особистої конкурентоспроможності. Причиною цього є дуже широке використання технології у додатках. Проте у даного підходу роботи над взаємодією БД та програмного коду, окрім великої кількості переваг, є й ряд недоліків, тому технологію потрібно використовувати обдуманно.

Список використаних джерел

1. <https://coderlessons.com/articles/java/5-prichin-ispolzovat-jpa-hibernate>
2. [https://ru.bmstu.wiki/ORM_\(Object-Relational_Mapping\)#.D0.94.D0.BE.D1.81.D1.82.D0.BE.D0.B8.D0.BD.D1.81.D1.82.D0.B2.D0.B0](https://ru.bmstu.wiki/ORM_(Object-Relational_Mapping)#.D0.94.D0.BE.D1.81.D1.82.D0.BE.D0.B8.D0.BD.D1.81.D1.82.D0.B2.D0.B0)
3. <https://habr.com/ru/post/667078/>

УДК 004.056.53:159.9

*Діденко М.М., здобувачка 4 курсу
спеціальності 125 «Кібербезпека»
Крижановський В.Г., д.т.н., професор
кафедри інформаційних технологій*

ПСИХОЛОГІЧНІ РИСИ ЛЮДИНИ, ЯКІ СПРИЯЮТЬ ЇЇ ВІДГУКУ НА ФІШИНГ

Донецький національний університет імені Василя Стуса, м. Вінниця

Чим швидше людство розвивається, чим більше новітніх пристроїв створюється – тим більше з'являється різноманітної цікавої інформації, до якої не кожна людина може мати доступ. Саме через це все частіше з'являються нові варіанти соціальної інженерії. Один з найпопулярніших методів являється фішинг.

Фішинг – техніка інтернет-шахрайства, що є спрямована на отримання конфіденційної інформації користувачів. Головним видом фітінгових атак є підроблений лист, відправлений жертві по електронній пошті, який виглядає як лист від офіційної структури (платіжної системи, банку, компанії, тощо). У листі міститься форма для введення персональних даних (пін-код, логін і пароль і т.п.) або посилання на web-сторінки, де розташовується така форма [1-2]. Коли користувач вводить той же пароль і логін, щоб отримати доступ до «псевдо-сайту», шахрай отримує доступ до даних користувача, після чого може користуватись ними як забажає, в той час, як користувач може навіть і не підозрювати.

З метою захисту своїх користувачів від фішингу виробники основних інтернет-браузерів домовилися про застосування однакових способів інформування людей про те, що той чи той сайт є підозрілим та може належати шахраям. Найновіші версії браузерів уже мають функцію «Антифішинг». Не дивлячись на всі існуючі наразі застережні заходи величезна кількість сучасних інтернет-користувачів та користувачів соцмережами, месенджерами продовжує «вестися» на різноманітні підступні фішингові прийоми та стратегії