ObjectPlusV2 – wykład MAS 04

Wojciech Fabjańczuk

Treść zadania

- Ekstensja w ramach tej samej klasy

- Jak ulepszyć ObjectPlus?

```java
public abstract class ObjectPlus implements Serializable {
    private static Map<Class, List<ObjectPlus>> allExtents = new Hashtable<>();

    // ...

    public static void writeExtents(ObjectOutputStream stream) throws IOException {
        stream.writeObject(allExtents);
    }

    public static void readExtents(ObjectInputStream stream) throws IOException, ClassNotFoundException {
        allExtents = (Hashtable) stream.readObject();
    }

    // ...
}
```

Cel: trwałość pól klasowych

- Pola **static** w podklasach nie są zapisywane

```java
public class Movie2_V2 extends ObjectPlusV2 implements Serializable {
    /* Class body */

    //    Business implementation

    private static String movieReviewSource = "Filmweb.pl";
    private static Date lastInstanceCreationDate = null;

    // ...
```

- Potrzebne: **struktura danych, szukanie podklas, filtrowanie pól, odczyt i zapis**

Przechowywanie pól klasowych

- Analogicznie:

```java
public abstract class ObjectPlusV2 implements Serializable {
    private static Map<Class, List<ObjectPlusV2>> allExtents = new Hashtable<>();
    private static Map<Class, Map<String, Object>> allStaticFields = new Hashtable<>();
```

Szukanie podklas (wszystkich)

- Zewnętrzna biblioteka **org.reflections**

- Classmap scanner

```java
private static Set<Class<? extends ObjectPlusV2>> getSubclasses() {
    Reflections reflections = new Reflections(
            prefix: "", new SubTypesScanner());
    return reflections.getSubTypesOf(ObjectPlusV2.class);
}
```

Szukanie podklas (z instancjami)

- Pomijamy podklasy bez instancji

```java
public abstract class ObjectPlusV2 implements Serializable {
    private static Map<Class, List<ObjectPlusV2>> allExtents = new Hashtable<>();

    private static Set<Class> getSubclasses() {
        return allExtents.keySet();
    }
}
```

Pobieranie pól

- Niezależnie od szukania podklas

```java
private static void readAllStaticFieldsFromSubclass(Class<? extends ObjectPlusV2> subclass)
        throws IllegalAccessException {
    LinkedHashMap<String, Object> staticFieldMap = new LinkedHashMap<>();

    for (Field field : subclass.getDeclaredFields()) {
        if (Modifier.isStatic(field.getModifiers())) {
            field.setAccessible(true);
            staticFieldMap.put(field.getName(), field.get(null));
        }
    }

    allStaticFields.put(subclass, staticFieldMap);
}
```

Odczyt i zapis

```java
public static void readSerializedSubclasses(ObjectInputStream stream)
        throws IOException, ClassNotFoundException, NoSuchFieldException, IllegalAccessException
    allExtents = (Hashtable) stream.readObject();
    allStaticFields = (Hashtable) stream.readObject();
    writeAllStaticFieldsToSubclasses();
}


public static void writeSerializedSubclasses(ObjectOutputStream stream)
        throws IOException, IllegalAccessException {
    readAllStaticFieldsFromSubclasses();
    stream.writeObject(allExtents);
    stream.writeObject(allStaticFields);
}
```

Przykład działania

```
1. Movie2_V2 Just after serializing:

---- Extent of the class: Movie2_V2 ----
Movie: Terminator 1, id: mt.mas.Movie2_V2@3712b94
Movie: Terminator 2, id: mt.mas.Movie2_V2@e320068
---- Static fields of the class: Movie2_V2 ----
movieReviewSource: Filmweb.pl
lastInstanceCreationDate: Tue Mar 30 21:10:55 CEST 2021

2. Movie2_V2 After temporary setting static fields to null:

---- Extent of the class: Movie2_V2 ----
Movie: Terminator 1, id: mt.mas.Movie2_V2@3712b94
Movie: Terminator 2, id: mt.mas.Movie2_V2@e320068
---- Static fields of the class: Movie2_V2 ----
movieReviewSource: (null)
lastInstanceCreationDate: (null)

3. Movie2_V2 After reading serialized subclasses:

---- Extent of the class: Movie2_V2 ----
Movie: Terminator 1, id: mt.mas.Movie2_V2@16f7c8c1
Movie: Terminator 2, id: mt.mas.Movie2_V2@2f0a87b3
---- Static fields of the class: Movie2_V2 ----
movieReviewSource: Filmweb.pl
lastInstanceCreationDate: Tue Mar 30 21:10:55 CEST 2021
```

Pokaz kodu

- https://github.com/wfabjanczuk/mas-04_static-serializing

- Branch **main** – wszystkie podklasy

- Branch **feature/without_dependencies** – tylko podklasy z instancjami