Міністерство освіти і науки України  
НТУУ «КПІ ім. І. Сікорського»

Кафедра автоматизації проектування енергетичних процесів і систем

Лабораторна робота №6

«Серіалізація»

Виконав: студент II курсу ТЕФ, групи TI-92

Сопронюк Ю. А.

Перевірив: Касьянов А. С.

Дата виконання: 25.12.2020

Київ-2020

**Варіант №11**

ЗАВДАННЯ

Доповнити класи із лабораторної роботи №3 серіалізацією. Впровадити інтерфейс *Serializable* для класів із асоціацією/агрегацією з іншими класами (поле або масив іншого вашого типу).

Хоча б для одного класу передбачити явно визначену серіалізацію - передбачити *transient* (несеріалізуємі) поля, які будуть записуватися як певний атрибут іншого (базового) типу у визначеному методі *writeObject*. Такі несеріалізуємі об’єкти повинні генеруватися у класі під час виконання програми при зчитуванні такого атрибуту у визначеному методі *readObject*. Записати об’єкти у файл і зчитати їх із файлу у програмі.

**ВИКОНАНАННЯ РОБОТИ**

**Код програми**

**Усі необхідні файли проекту можна отримати за посиланням (ветка lab6):**

*https://github.com/steadyGuy/oop\_labs/tree/lab6*

Крім основних файлів з розширенням *.java*, тут також знаходиться звіт у форматі *.docx*. *Main.java* – точка входу у програму.

**Результат роботи програми**

**

**Згенерований файл**

**

Оскільки створюється .*jar* файл для компілювання, то шлях для створення *.bin* файлу був взятий абсолютним, відносно шляху компілювання.

**ВИСНОВОК**

На даній лабораторній роботі було доповнено вже готову програму (предметна область “*туристичні набори*” з 11 варіанту) серіалізацією. Насамперед, для певних класів був імплементований інтерфейс *Serializable*, який дозволяє серіалізувати екземпляри класів з цим інтерфейсом. Сама ж серіалізація – це процес, який переводить об'єкт в послідовність байтів, по яким потім його можна повністю відновити. Тобто її сутність дозволяє ніби «дати життя» об'єкту також між запусками програми (адже як відомо, життєвий цикл програми – це її запуск і завершення). Після того, як необхідні об’єкти були серіалізовані, їх можна легко зчитати і записати за допомогою основних методів для роботи з потоками в java (для прикладу, одні з основних *writeObject* та *readObject* – запис та зчитування даних з потоку відповідно).

Також, було враховно ті об’єкти – які не повинні серіалізуватися. Вони були помічені спеціальним ключовим словом *transient* (в моєму випадку, це був екземпляр класа *StoreInfo*). Однак, сам об’єкт може генеруватися під час виконання програми, адже за допомогою методів *writeObject* та *readObject*, які були впроваджені в класі *TravelStore* – можна записати певний примітивний атрибут, а потім, на його основі при десереалізації – згенерувати новий.