

Лабораторна робота № 6

Технологія серіалізації у Java

Мета роботи: Ознайомитись з технологією серіалізації Java та навчитися виконувати серіалізацію та десеріалізацію об'єктів.

1. Завдання

1. Оберіть завдання у наведеній нижче таблиці відповідно до номеру Вашого варіанта (зазначений у списку варіантів індивідуальних завдань лабораторних робіт на сторінці дисципліни у Moodle).

V	Завдання	V	Завдання
1	Створіть клас із приватними полями. Виконайте серіалізацію та десеріалізацію об'єкта, щоб перевірити, чи збереглися значення приватних полів. Наведіть вміст файлу серіалізації.	8	Створіть клас, який не реалізує <code>Serializable</code> , але його об'єкт використовується у класі, що серіалізується. Використайте модифікатор <code>transient</code> , щоб уникнути помилки. Наведіть вміст файлу серіалізації.
2	Додайте до довільного класу поле з модифікатором <code>transient</code> . Перевірте, чи зберігається його значення після десеріалізації. Наведіть вміст файлу серіалізації.	9	Реалізуйте програму, яка генерує та обробляє винятки, що виникають під час серіалізації або десеріалізації (наприклад, через відсутність файлу або невідповідність класів).
3	Створіть клас, який містить список об'єктів іншого класу. Реалізуйте серіалізацію та десеріалізацію цього класу. Наведіть вміст файлу серіалізації.	10	Реалізуйте серіалізацію об'єкта через мережевий потік (використовуючи сокет) між двома програмами.
4	Виконайте серіалізацію об'єкта класу у файл. Потім змініть структуру класу (наприклад, додайте нове поле) та спробуйте виконати десеріалізацію. Додайте поле <code>serialVersionUID</code> для вирішення проблеми. Наведіть вміст файлу серіалізації.	11	Створіть серіалізований клас, який використовує бібліотеку <code>GSON</code> або <code>Jackson</code> , і перетворіть об'єкт на <code>JSON</code> .
5	Створіть клас із приватним полем <code>password</code> . Використовуйте методи <code>writeObject</code> і <code>readObject</code> , щоб серіалізувати об'єкт, але приховати значення пароля. Наведіть вміст файлу серіалізації.	12	Створіть клас <code>User</code> із полями <code>username</code> та <code>password</code> . Реалізуйте інтерфейс <code>Externalizable</code> і створіть методи <code>writeExternal</code> та <code>readExternal</code> , щоб серіалізувати лише ім'я користувача, але не пароль.
6	Створіть суперклас і підклас. Реалізуйте серіалізацію об'єкта підкласу. Перевірте, чи зберігаються поля суперкласу. Наведіть вміст файлу серіалізації.	13	Створіть список об'єктів класу <code>Book</code> із полями <code>title</code> та <code>author</code> . Використайте бібліотеку Jackson або JAXB , щоб серіалізувати список у XML-файл, а потім десеріалізувати його назад.
7	Створіть два класи, де один містить посилання на інший. Виконайте серіалізацію та десеріалізацію об'єкта	14	Створіть клас із вкладеним об'єктом, що реалізує <code>Externalizable</code> . Перевірте, як працює серіалізація об'єкта

V	Завдання	V	Завдання
	основного класу. Наведіть вміст файлу серіалізації.		основного класу разом із вкладеним об'єктом.

- Наведіть код програми та вміст файлу серіалізації у шістнадцятковій системі, якщо вказано у завданні, до звіту та опишіть ці поля.
- Зробіть висновок про застосування та особливості використання технології серіалізації.

2. Контрольні питання

- Що таке серіалізація у Java і для чого вона використовується?
- Який інтерфейс потрібно реалізувати, щоб клас став серіалізованим?
- Які вимоги до класу, щоб він міг бути серіалізованим?
- Чим серіалізація відрізняється від десеріалізації?
- Що таке `serialVersionUID` і для чого він використовується?
- Які дані не серіалізуються за замовчуванням?
- Що відбувається, якщо клас має поле з модифікатором `transient`?
- Чи можна серіалізувати клас із полями, які не реалізують `Serializable`?
- Як серіалізувати об'єкти, що містять посилання на інші об'єкти?
- Що відбудеться, якщо при десеріалізації версія класу відрізняється від серіалізованої?
- Чим інтерфейс `Externalizable` відрізняється від `Serializable`?
- Які методи потрібно реалізувати в класі, що використовує `Externalizable`?
- Як реалізувати кастомну серіалізацію з використанням `Externalizable`?
- Що станеться, якщо у методі `writeExternal` забути записати якесь поле?
- У яких випадках використання `Externalizable` може бути доцільним?
- Як серіалізувати об'єкт у файл і прочитати його назад?
- Як серіалізувати об'єкт у масив байтів?
- Які винятки можуть виникати під час серіалізації та десеріалізації?
- Що таке глибока копія об'єкта і як її реалізувати за допомогою серіалізації?
- Як серіалізувати об'єкт, який містить статичні поля?
- Чи можна серіалізувати об'єкти через мережу? Як це зробити?
- Які бібліотеки використовуються для серіалізації у XML?
- Як серіалізувати об'єкт у формат JSON? Наведіть приклад.
- Які переваги та недоліки серіалізації у порівнянні з іншими способами збереження даних?
- Як серіалізація впливає на продуктивність програми?

3. Література

- Методичні матеріали курсу "Технологія Java (Advanced)" URL: <https://moodle.znu.edu.ua/course/view.php?id=16444> (дата звернення 30.08.2024).

2. Schildt H., Coward D. Java™. The Complete Reference. Thirteen Edition, McGraw Hill, 2024. 2750 p.
3. Horstmann C. S. Core Java, Volume I: Fundamentals. 12-th Ed. "Addison-Wesley", 2022. 1197 p.