



ЧАСТИНА 3

Розширені технології

JAVA

**за вимогами корпоративного іспиту
*1Z0-809 “Oracle Certified Professional,
Java SE8 Programmer”***

Java Camp 2022

Розширені технології Java

Програма навчання

Модуль 1. Алгоритми та контейнери

- 1.Опрацювання текстових даних (*parsing*) на основі технології *regex*.
- 2.Серіалізація файлів. Інтерфейси *Serializable* та *Externalizable*. Особливості композиції та наслідування при серіалізації.
- 3.Фреймворк колекцій. Принципи побудови *hash*-таблиць. Інтерфейси *Set*, *Queue*, *Map*, *Iterator*. Класи *TreeSet*, *PriorityQueue*, *Deque*, *TreeMap*, *HashMap*.
- 4.Робота з базами даних. Інтерфейс *JDBC*.
- 5.Побудова та використання *generic*-класів та *generic*-методів. Особливості поліморфізму при використанні *generic*-колекцій.

Модуль 2. Потокове програмування

- 6.Основи створення та використання потоків. Клас *Thread*. Інтерфейси *Runnable*, *Callable*, *Executor*, *ExecutorService*.
- 7.Основи синхронізації потоків. Ключове слово *synchronized*. Синхронізація статичних методів. Пакет *java.util.concurrent.atomic*.
- 8.Реалізація синхронізації на основі ресурсів пакету *java.util.concurrent*. Потокобезпечні колекції.
- 9.Реалізація синхронізації на основі ресурсів пакету *java.util.concurrent.lock*.
- 10.Взаємодія потоків. Методи *wait*, *notify* класу *Object*. Поняття взаємного блокування (*deadlock*).
- 11.Завершальне завдання по *Threads – Airport*.
- 12.Проміжні та термінальні методи інтерфейсу *Stream* для формування та опрацювання потоків.
- 13.Клас *Optional*. Особливості технології потокового зведення. Метод *reduce*.
- 14.Методи класу *Collectors*.
- 15.Розпаралелювання потоків. Порівняльний аналіз послідовних та паралельних потоків, визначення доцільності розпаралелювання потоків.
- 16.Завершальне завдання.

Тема 1. Опрацювання текстових даних (parsing) на основі технології regex.

Відео: 11_Parsing.

Література: [Хорстманн, т.2] – с.148...161,
[Шилдт] – с. 727...729, 1153...1163,
[ukrSierra6] - 11_Parsing.

Тести обов'язкові (12): [Sierra5] – ch.6 – 1, 14,
[Sierra6] – ch.6 – 14, 15.
[SierraExam] – PracticeExam1 – 41, 53,
PracticeExam2 – 14,
PracticeExam3 – 21, 48,
PracticeExam4 – 21,
[Sierra7] – ch.8 – 3, 4,

Тести додаткові (15): [Raposa] – AssTest – 29,
ch.4 – 21...24,
[Sanghera] - PracticeExam – 45,
[Heller] – PracticeExam – 26,
[Ganesh] – ch.7 – 2, 4...6,
MockExam1 - 57,
MockExam2 – 20, 28, 80.

Програмне завдання. За допомогою класів Pattern, Matcher / Scanner (за вибором слухача) в заданому текстовому файлі видалити всі слова заданої довжини, що починаються на приголосну букву. Опрацьовані фрагменти тексту заносити до нового текстового файлу безпосередньо після опрацювання без утворення буферних колекцій або без перетворення всього файлу у текст.

Тема 2. Серіалізація файлів. Інтерфейси Serializable та Externalizable. Особливості композиції та наслідування при серіалізації.

Відео: 2_Serialization

Література обов'язкова: [ukrSierra6] – 12_Serialization,
[Шилдт] – с.826...828, 842...849,

Тема, відсутня у відео – інтерфейс Externalizable: [Эккель] – с.540...548,

Література додаткова: [Хорстманн, т.2] – с.102...119,
[Вязовик] – с.525...541.

Тести обов'язкові (9): [Sierra5] - ch.6 - 2, 6, 15,
[SierraExam] - PracticeExam2 – 11,
[Ganesh] - MockExam1 – 66,
MockExam2 – 13,
[Raposa] - ch.4 – 14, 16,
[Sanghera] - ch.8 – 10.

Тести додаткові(19): [Ganesh] - PreTest – 29,
MockExam1 – 65,
[Sierra6] - ch.6 - 12,
[Raposa] - AssTest – 16,
[Sanghera] - ch.8 – 9,
PracticeExam – 35,
[Heller] - AssTest – 3, 30,
ch.9 – 11...15, 18,
PracticeExam – 21...23,
[Jaworsky] - p.282 – 1, 15,
[Green] - 8, 23.

Програмне завдання. Дано наступних 7 класів:

- “Транспортний засіб” з полями:
 - швидкість,
 - рік виробництва,
 - двигун,
- “Літак” (наслідує “Транспортний засіб” та реалізує Serializable) з полями:
 - модель,
 - дальність польоту,
 - шасі,
- “Корабель” (наслідує “Транспортний засіб” і реалізує Externalizable) з полями:
 - водотонажність,
 - довжина,
 - шлюпка,
- “Двигун” з полями:
 - тип,
 - потужність,
- “Шасі” з полями:
 - колесо,
 - кількість коліс,
- “Шлюпка” з полями:
 - кількість пасажирів,
 - матеріал,
- “Колесо” з полями:
 - навантаження,
 - діаметр.

Необхідно:

- 1) створити об’єкти класів літаків та кораблів, які зберегти у двох відповідних ArrayLists, сортованих по роках виробництва,
- 2) серіалізувати створені ArrayLists до відповідних файлів: PLANES за допомогою інтерфейсу Serializable та SHIPS за допомогою інтерфейсу Externalizable,
- 3) за допомогою інтерфейсів Serializable/Externalizable десеріалізувати файли PLANES та SHIPS до двох відповідних сортованих ArrayLists.

Тема 3. Фреймворк колекцій. Принципи побудови hash-таблиць. Інтерфейси Set, Queue, Map, Iterator. Класи TreeSet, PriorityQueue, Dequeue, TreeMap, HashMap.

Відео: 3_TreeSets,
4_Queues_LinkedLists,
5_TreeMaps,
6_HashCode_HashMaps,
7_Iterators.

Література: [ukrSierra6] – 13_Collections,
[Шилдт] – с.636...637, 642...649, 654...661, 662...665, 672...684,
[Хорстманн] – с.437...476.

Тести обов'язкові (15): [Sierra5] – ch.7, № 1, 3, 5...8, 11, 12, 15,
[Sierra6] – ch.7, № 9...13, 15.

Тести додаткові(8): [Raposa] – ch.7, № 9...11, 13,
[Heller] – ch.8, № 12, 18,
[Jaworsky] – p.175, № 32, 50.

Програмне завдання. На площині задано N точок. Необхідно у файл LINES записати два HashMap:

- із ключами у вигляді об'єктів класу Point, що визначаються цілими координатами точок, та значеннями у вигляді кількості прямих, що проходять через цю точку та, щонайменше, ще через одну точку,
- із ключами у вигляді об'єктів класу Line, що визначаються параметрами K та B прямої $y = K \cdot x + B$, та значеннями у вигляді кількості точок, що належать цій прямій.

Тема 4. Робота з базами даних. Інтерфейс JDBC.

Відео: 24_JDBC.

Література: [Хорстманн], с.287...319.

Тести (15): [Sierra7] – ch.15, № 10...15,
[MalaGupta_OCP] – ch.9, № 4...9,
[Ganesh] – ch.10, № 5...7.

Програмне завдання. Прочитати та занести дані у відповідні колекції дані з існуючої бази у форматі:

N	Bottle	Volume	Material
1	Wine	0.75	Glass
2	Juice	0.25	Metal
...			

Завдання для формування запитів:

- зчитувати пляшки місткістю: не більше 0.5 л, в межах від 0.51л до 0.99 л, не меншою за 1.0 л,
- зчитувати пляшки для: вина, соків, води,
- зчитувати пляшки, зроблені з: металу, скла, пластику.

Тема 5. Побудова та використання generic-класів та generic-методів. Особливості поліморфізму при використанні generic-колекцій.

Відео: 8_GenericBasic,
9_GenericMethods,
10_GenericLimited,
11_GenericPolymorphism.

Література: [ukrSierra6] – 14_Generics,
[Шилдт], с. 413...436, 437...452, 454...458.
[Хорстманн], с.393...417,
[Эккель], с.397...402, 404...408.

Тести (11): [Sierra5] – ch.7, № 10, 14, 16,
[Sierra6] – ch.7, № 14,
[Ganesh] – ch.6, № 4,
MockExam1 – 19, 28, 33, 35, 36,
MockExam2 – 11.

Програмне завдання. Створити два generic-методи для пошуку спільних елементів в двох контейнерах різних типів (ArrayList та одновимірний масив) з даними одного типу без дозволу явного наслідування. В першому методі повернути Map із спільними елементами в якості ключів та повною кількістю повторів елементів в якості значень. В другому методі до ArrayList додати всі елементи з масиву, що є відсутніми в ArrayList. Випробувати методи на об'єктах одного класу (наприклад, Animal) із набором полів за вибором студента. Проаналізувати можливість використання створених методів для передавання в них в якості параметрів контейнерів, що можуть містити об'єкти підкласів (Dog, Cat, Poodle, тощо). При визначенні спільних елементів в об'єктах підкласів використовувати тільки поля суперкласу.

Тема 6. Основи створення та використання потоків. Клас Thread. Інтерфейси Runnable, Callable, Executor, ExecutorService.

Відео: 12_ThreadBasic
13_CallableExecutors

Література: [ukrSierra6] – 15_ThreadBasic,
[Sierra6] – ch.9, fig.9-1, 9-2,
[Шилдт] – ст.303...319, 333...335, 1088...1094,
[Хорстманн] - с. 661...676, 722...732.

Тести обов'язкові (22): [SierraExam] – AssTest1, # 3, 12,
AssTest2, # 8,
PE2, # 30, 48,

PE3, # 8, 31, 42, 52,
PE4, # 11, 48,
[Sierra5] - ch.9, # 1,
[Sierra7] - ch.13, # 13, 14,
 ch.14, # 11, 12,
[Ganesh7] – ch.13, # 2,
[Jaworsky] - p.140, # 28,
[Raposa] - ch.5, # 2...5.

Програмне завдання. Створити багатопотокову систему для визначення кількості розділових знаків в трьох текстових файлах по кожному файлу окремо. Для кожного файлу за власним вибором обрати свій спосіб створення потоку (за допомогою інтерфейсів Runnable, Callable, Executor, ExecutorService). НЕ ВИКОРИСТОВУВАТИ для прочитаного тексту створення проміжних колекцій або цілий String. Порівняти часові витрати при вирішенні завдання одно- та багатопотоковим способами. СТРОГО ПІСЛЯ завершення читання даних з файлів для збереження результатів створити TreeSet, посортований по кількості розділових знаків у файлах від більшого значення до меншого, (використати обгортку або підклас класу File).

Тема 7. Основи синхронізації потоків. Ключове слово synchronized. Синхронізація статичних методів. Пакет java.util.concurrent.atomic.

Відео: 14_Synchronization,
15_Atomic.

Література: [ukrSierra6] – 16_Synchronization,
[Хорстманн] - с. 676...681, 689...694, 697...699,
[Шилдт] - с. 319...323, 1099...1101.

Тести (16): [SierraExam] – AssTest2, # 14,
PE1, # 42, 60,
PE2, # 16,
PE3, # 25,
PE4, # 28, 55,
[Sierra7] - ch.13, # 11, 15...17,
 ch.14, # 7,
[Raposa] - ch.5, # 9, 10, 15.

Програмні завдання.

1. Із використанням ключового слова synchronized створити синхронну багатопотокову систему для визначення загальної суми всіх наявних цілих та дробових

чисел в трьох текстових файлах. НЕ УТВОРЮВАТИ суму всіх чисел з кожного файлу окремо, потім додаючи три суми, результат треба утворювати послідовно - по мірі зчитування чисел з файлів. НЕ ВИКОРИСТОВУВАТИ проміжне створення колекцій для прочитаного тексту і не перетворювати весь файл у текст.

2. Аналогічну систему створити із використанням ресурсів пакету `java.util.concurrent.atomic`.

Тема 8. Реалізація синхронізації на основі ресурсів пакетів `java.util.concurrent`. Потокобезпечні колекції.

Відео: 16_ConcurrentCollections

Література: [Шилдт] - с. 1095...1096,
[Хорстманн] - с.705...722.

Тести (5): [Sierra7] – ch.14, # 1...5.

Програмне завдання. Із використанням потокобезпечних колекцій пакету `java.util.concurrent` створити синхронну багатопотокову систему для формування з трьох текстових файлів колекції з кількістю повторень кожного слова **ДОВЖИНОЮ БІЛЬШЕ ОДНІЄЇ БУКВИ, В ЯКИХ СПІВПАДАЮТЬ ПЕРША ТА ОСТАННЯ БУКВИ**. НЕ ВИКОРИСТОВУВАТИ проміжне створення колекцій для прочитаного тексту.

Тема 9. Реалізація синхронізації на основі ресурсів пакету `java.util.concurrent.lock`.

Відео: 17_LocksAndVolatile

Література: [Хорстманн] - с.695...696,
[Шилдт] - с.1096...1099.

Тести (2): [Sierra7] – ch.14, # 8, 9.

Програмне завдання. Із використанням засобів ресурсів пакету `java.util.concurrent.locks` створити синхронну багатопотокову систему для визначення загальної суми всіх цифр в трьох текстових файлах. У випадку блокування потоку, він повинен виводити своє ім'я з маркою "LOCKED". НЕ ВИКОРИСТОВУВАТИ проміжне створення колекцій для прочитаного тексту.

Тема 10. Взаємодія потоків. Методи `wait`, `notify` класу `Object`. Поняття взаємного блокування (deadlock), запобігання deadlock.

Відео: 18_WaitNotify.

Література: [ukrSierra6] – 17_WaitNotify,

[Хорстманн] - с.699...702,
[Шилдт] - с.323...330,
[Эккель] - с.598...601, 602...607.

Тести (4): [Jaworsky] - p.141, # 53,
[Sierra7] - ch.13, # 6, 7, 12,

Програмне завдання. Спроектувати двопотокову систему обробки текстових файлів наступним чином: перший потік визначає кількість пробілів у файлі, якщо кількість пробілів є парною - другий потік робить у файлі прописними (великими) перші букви всіх слів, якщо непарною – останні букви. Забезпечити наступність обробки файлів потоками – обробка другим потоком поточного файлу здійснюється під час обробки першим потоком наступних файлів.

Порівняти ефективність роботи розробленої системи із класичною багатопотоковою системою, де кожний потік обробляє повністю один файл, а також у випадках обробки файлів приблизно однакового обсягу та з критично різними обсягами.

Тема 10. Завершальне завдання по Threads - Airport.

Пасажирів, прибувших на трьох (чотирьох, п'яти) літаках одночасно, розвозять мікроавтобуси. Завантаженість літаків – рівно 100 пасажирів. Мікроавтобуси мають місткість 6, або 7, або 8 пасажирів і їдуть до 4 різних міст. Завантаженість мікроавтобусів повинна бути 100%, за винятком мікроавтобусів з останніми пасажирами. Пасажири їдуть сім'ями від 1 до 4 осіб. Розділяти сім'ї по мікроавтобусах не можна.

Зробити спочатку однопотокову систему, потім – синхронну багатопотокову, із синхронізацією по літаках або по автобусах.

Три класи.

```
class Family{
    String name;    // twoLetters "aa", "ab", ..., "zz" – for example, up to 100 names
    String travelTo; // 4 cities – "Kalush", "Kosiv", "Galych", "Kolomiya"
    int count;      // family members count, from 1 to 4 members
}
class Plane{
    List<Family> families; // exactly 100 family members
    int id;               // exactly 1, 2, 3
}
class Bus{
    int passengersCount; // 6 or 7 or 8
    String driveTo;      // 4 cities – "Kalush", "Kosiv", "Galych", "Kolomiya"
}
```

Тема 12. Проміжні та термінальні методи інтерфейсу Stream для формування та опрацювання потоків.

Відео: 20_Stream1_LazyEagerOperations

Література: [Шилдт] - с. 1123...1132, 1138...1147,
[Хорстманн, т.2] - с. 19...34, 42...51,
[Уорбэртон] - с.31...42, 44...53, 70...72.

Тести (11): [Boyarsky] – AssT, # 11, 20,
ch.4, # 1...4, 7, 10, 11, 13, 17.

Програмні завдання.

На основі методів інтерфейсу Stream виконати наступні дії:

- в тексті реалізувати початок кожного слова з прописної (великої) букви;
- вивести всі речення тексту за зростанням кількості слів, припустимо, що всі речення завершуються тільки крапкою;
- колекцію цілих чисел поділити на колекції з додатніх та від'ємних елементів.

Тема 13. Клас Optional. Особливості технології потокового зведення. Метод reduce.

Відео: 21_Stream2_Optional_Reduce

Література: [Шилдт] - с. 1132...1135,
[Хорстманн, т.2] - с.35...42, 57...59,
[Уорбэртон] - с.43...44.

Програмні завдання.

На основі методів потокового зведення інтерфейсу Stream виконати дії:

- утворити третій контейнер ArrayList як конкатенацію двох масивів різної довжини;
- визначити середнє арифметичне елементів масиву цілих чисел та кількість елементів, що є більшими за середнє арифметичне;
- з колекції цілих чисел видалити дублікати максимуму та мінімуму.

Тема 14. Методи класу *Collectors*.

Відео: 22_Collectors_1, 22_Collectors_2, 22_Collectors_3,

Література: [Хорстманн, т.2] - с. 42...57,
[Уорбэртон] - с.75...96.
[Шилдт] - с. 1143...1147,
<https://annimon.com/article/2778#collectors-methods> - р.7 “Методы Collectors”

Програмні завдання. На основі методів класу `Collectors` виконати дії:

- в колекції цілих чисел поміняти місцями максимум та мінімум;
- визначити середнє арифметичне елементів колекції цілих чисел та сформувати вихідну колекцію з елементів, що є більшими за середнє арифметичне;
- в кожному реченні тексту без використання попереднього розбиття на речення визначити різницю між кількістю приголосних та голосних букв і сформувати відповідний `Map` (key: номер речення, value: визначена різниця між кількістю приголосних та голосних букв).

Тема 15. Розпаралелювання потоків. Порівняльний аналіз послідовних та паралельних потоків, визначення доцільності розпаралелювання потоків.

Відео: 23_`ParallelStreams`

Література: [Шилдт] - с. 1135...1138,
[Хорстманн, т.2] - с. 65...70,
[Уорбэртон] - с.98...113.

Програмне завдання. Із використанням паралельних потоків в масиві цілих чисел знайти парні числа, з яких утворити посортований `Map` (key: знайдене парне число, value: кількість його повторів в масиві). Порівняти часові витрати на рішення завдання послідовним та паралельними потоками.

Тема 16. ЗАВЕРШАЛЬНЕ ЗАВДАННЯ.

Із комплексним використанням технологій багатопотоковості (`threads`) та паралельних потоків фреймворку `API Streams` створити максимально паралелізовану синхронну систему для визначення середньої довжини всіх слів в `N` текстових файлах.

ЛІТЕРАТУРА

1. Шилдт Герберт. Java. Полное руководство. 10-е изд.: Пер. с англ. – СПб.: ООО “Альфа-книга”, 2018.- 1488 с.
2. Вязовик Н.А. Программирование на Java. - М.: ИНТУИТ, 2016.
3. Іщеряков С.М. Комп’ютерна схемотехніка. – Івано-Франківськ: Полум’я, 2011. – 114 с.
4. Хорстманн Кей С. Java. Библиотека профессионала. 11-е изд.: – СПб.: Диалектика, 2019. Т.1. Основы. - 864 с
5. S.G.Ganesh, Tushar Sharma. Java SE7 Programmer Exams 1Z0-804 and 1Z0-805. A Comprehensive OCPJP 7 Certification Guide.– APRESS, New York, 2012.-644 p.
6. Philip Heller, Simon Roberts. Complete Java 2 Certification Study Guide. 5-th edition. – SYBEX, Inc., Alameda, 2005.- 560 p.
7. Raposa, Richard F. SCJP Sun Certified Programmer for Java Platform, SE6, Study Guide. – Wiley Publishing, Inc., Indianapolis, 2009.- 583 p.

8. Khalid A.Mughal, Rolf W.Rasmussen. A Programmer's Guide to Java SE8 Oracle Certified Associated (OCA). – Addison-Wesley Publishing, 2017.
9. Kathy Sierra, Bert Bates. SCJP Sun Certified Programmer for Java 5 Study Guide. – McGraw Hill Osborne, New York, 2006.- 924 p.
- 10.Jamie Jaworsky. Java 2 Certification Training Guide. – New Riders Publishing, Indianapolis, 1999.- 399 p.
- 11.Mala Gupta. OCA Java SE7 Programmer I Certification Guide. Prepare for the 1Z0-803Exam. – Manning, Shelter Island, 2013.- 528 p.
- 12.Paul Sanghera. SCJP Exam for J2SE5. A Concise and Comprehensive Study Guide for the Sun Certified Java Programmer Exam. – APRESS, New York,2006.- 423 p.
- 13.Jeanne Boyarsky, Scott Selikoff. OCP Oracle Certified Professional Java SE8 Programmer II. Study Guide. Exam 1Z0-809- John Wiley & Sons, Inc., Indianapolis, 2016.- 669 p.
- 14.Richard M.Reese. Oracle Certified Associate Java SE7 Programmer Study Guide. – Packt Publishing, Birmingham, 2012.- 313 p.
- 15.Kathy Sierra, Bert Bates. OCA/OCP Java SE7 Programmer I & II Study Guide (Exams 1Z0-803 & 1Z0-804) (Oracle Press).– McGraw Hill Osborne,New York,2015.-1088 p.

Програму розроблено ТОВ “Інновейшн Девелопмент Фаундейшн”