

## ЛАБОРАТОРНА РОБОТА 3

Тема: Класи в Java. ООП та колекції.

Мета: ознайомитися з основами можливостями об'єктно-орієнтованого програмування у Java. Навчитися використовувати колекції та виключення при написанні прикладних програм.

### **Теоретичні питання**

1. Визначення класу в Java. Модифікатори способу доступу.
2. Змінні класу і змінні реалізації. Використання модифікаторів `static` і `final` для змінних у класі.
3. Методи класу. Статичні і фінальні методи. Конструктори. Метод `finalize()`.
4. Абстрактні методи і класи. Інтерфейси.
5. Клас `Object` і його методи.
6. Рефлексія. Клас `Class` і оператор `instanceof`.
7. Ієрархія класів стандартної бібліотеки Java. Класи `System`, `Math`, `String`, `java.util.Arrays`.
8. Класи-колекції в Java. Інтерфейси `List`, `Set` і `Map`. Клас `java.util.Collections`.
9. Виключення. Класи виключень. Генерація та обробка виключень.

### **Контрольні питання**

1. Нехай оголошено рядок `String s = new String("aaa");`. Який результат дадуть вираження `(S=="aaa")`, `(S.equals("aaa"))` і чому?
2. Нехай у класі програми оголошені методи:

```
static void change (String s) { s = "bbb";}
static void change (int a []) {
    if (a != null) for (int i = 0; i <a.length; i ++) a [i] = 0;
}
static void change (int a [] []) {
    if (a != null)
        for (int i = 0; i <a.length; i ++)
            for (int j = 0; j <a [i] .length; j ++) a [i] [j] = 0;
}
```

а також локальні змінні:

```
String s = "aaa";  
int x [] = {1,2,3}, y [] [] = {{1,2,3}, {4,5,6}, {7,8,9}};
```

Чи зміняться значення цих змінних після кожного з таких викликів:

а) `change(s.clone()); change(x.clone()); change(y.clone());`

б) `change(s); change(x); change(y);`

і чому?

3. Чим відрізняється дія модифікаторів способу доступу `public`, `private`, `protected` у Java від іншої знайомої вам мови (C++, C# або ін.)?
4. Опишіть дію модифікаторів `static` і `final` при оголошенні змінних у класі. Наведіть приклади використання таких змінних.
5. Опишіть дію модифікаторів `static` і `final` при оголошенні методів класу. Наведіть приклади використання таких методів.
6. Чим конструктор відрізняється від інших методів класу?
7. Для чого використовуються ключові слова `this` і `super`? Наведіть приклади їх використання.
8. Як оголошуються і для чого використовуються абстрактні методи і класи, а також інтерфейси?
9. Опишіть призначення методів класу `Object`. У яких випадках необхідно перевизначати кожен з цих методів?
10. Які можливості надає інтерфейс `Comparable`? Як він реалізується?
11. Коротко опишіть призначення та основні можливості класів `System`, `Math`, `String`, `java.util.Arrays`.
12. Які реалізації мають інтерфейси `List`, `Set` і `Map`, і в чому відмінність цих реалізацій?
13. Які можливості надає клас `java.util.Collections`?
14. Як генеруються і обробляються виключення в Java?

### **Індивідуальні завдання**

Для виконання кожного індивідуального завдання напишіть консольний додаток. Реалізуйте введення вихідних даних із клавіатури і контроль

правильності введення. Результат виведіть у консоль у відформатованому і зручному для читання вигляді. Реалізуйте обробку можливих виключень.

### Загальне завдання

1. Напишіть програму, яка демонструє різні способи копіювання і порівняння масивів. Для цього створіть п'ять масивів рядків:

```
String a [] = { "1", "2", "3"}, b1 [] = a;  
String b2 [] = a.clone(),  
        b3 [] = Arrays.copyOf (a, a.length),  
        b4 [] = { "1", "2", "3"};
```

Попарно порівняйте ці масиви за допомогою операції порівняння (==), методу Object.equals(), методу Arrays.equals() і методу Arrays.deepEquals(). Результати порівняння виведіть у вигляді наступної таблиці:

b	b==a	b.equals(a)	Arrays.equals(a,b)	Arrays.deepEquals(a,b)	b[0]==a[0]
b=a	true	true	?	?	?
b=a.clone()	?	?	?	?	?
b=Arrays.copyOf(a)	?	?	?	?	?
b4={"1","2","3"}	?	?	?	?	?

Виведення одного рядка таблиці реалізуйте у вигляді окремого методу.

Поясніть отримані результати.

2. Виконайте завдання 1 для двовимірних масивів рядків. Поясніть результати.
3. Виконайте завдання 3 за варіантом, при цьому для всіх варіантів необхідно:
  - у базовому класі визначити конструктор за замовчуванням, конструктор із параметрами, а також конструктор копіювання;
  - у базовому класі перевизначити всі нефінальні методи класу Object;
  - кілька об'єктів обох класів зберегти в один масив;

- пройти за масивом і для кожного об'єкта вивести:
    - ім'я класу цього об'єкта;
    - результати роботи методів, визначених у базовому класі;
    - результати роботи методів, визначених тільки в похідному класі (якщо це можливо для цього об'єкта).
4. Об'єднайте кілька створених об'єктів свого класу (з завдання 3 за *варіантом*) у список List. Реалізуйте додавання, видалення елементів зі списку, а також виведення списку на екран. Для отриманого списку виконайте *завдання 4 за варіантом* двома способами:
- а) за допомогою методів класу Collections;
  - б) без використання методів класу Collections (тобто реалізувавши алгоритм самостійно).
- 5\*. (Для програмістів) Збережіть ті ж об'єкти в колекцію HashMap, де ключами будуть строкові імена об'єктів. Напишіть методи для обробки такої колекції:
- метод для пошуку значення за ключем;
  - метод для пошуку ключів за значенням;
  - метод, що виконує операцію з *завдання 5 за варіантом*.

### **Варіант 1**

3. Створіть клас «Автобусний маршрут». Внутрішнє представлення маршруту – список (List) або масив назв зупинок (пунктів, якими проходить маршрут), а також масив інтервалів за часом між сусідніми зупинками. Реалізуйте методи класу (статичні) і методи реалізації (нестатичні):
- для додавання нових зупинок;
  - для виведення маршруту в рядок у вигляді «Початковий\_пункт – пункт1 – Пункт2 ... Кінцевий\_пункт».

Створіть похідний клас, у якому:

- додайте метод для знаходження часу проїзду між будь-якими двома заданими зупинками;
- перевизначте метод `toString()` для виведення маршруту у вигляді «Маршрут від Початковий\_пункт до Кінцевий\_пункт займає X хвилин».

4. Напишіть метод для сортування списку маршрутів за назвою початкового пункту, а потім за назвою кінцевого пункту.
5. Напишіть метод, що приймає як параметри дві колекції `Map` і видаляє з першої колекції всі значення (*не ключі*), які зустрічаються у другій колекції.

### **Варіант 2**

3. Створіть клас «Ламана лінія». Внутрішнє представлення лінії – список (`List`) або масив цілочисельних координат її вузлових точок. Метод `toString()` повинен виводити лінію у вигляді «лінія з точки  $(x_1, y_1)$  до точки  $(x_n, y_n)$ ». Реалізуйте методи класу (статичні) і методи реалізації (нестатичні):
  - для додавання і видалення вузлів;
  - для знаходження загальної довжини лінії;
  - для виведення в рядок послідовності вузлових точок лінії.

Створіть похідний клас, у якому:

- перевизначте метод `toString()` для виведення лінії у вигляді: «лінія довжини  $x$  з точки  $(x_1, y_1)$  до точки  $(x_n, y_n)$ »;
- додайте метод перевірки, чи має лінія повторювані вузли.

4. Напишіть метод для сортування списку ліній за їх довжиною. Порядок сортування – від довгих ліній до коротких.
5. Напишіть метод, що приймає як параметри дві колекції `Map` і створює нову колекцію `Map`, у яку потрапляють тільки ті ключі, які присутні в першій колекції і відсутні у другій.

### Варіант 3

3. Створіть клас «Колір». Внутрішнє представлення кольору – три цілі числа в діапазоні [0; 100], що задають відсоток яскравості червоної, зеленої і синьої складових. Реалізуйте методи класу (статичні) і методи реалізації (нестатичні):

- для створення кольору за заданими трьома цілими числами в діапазоні [0; 255];
- для створення кольору за трьома дійсними числами в діапазоні [0; 100] або [0; 1];
- метод, що за заданими двома кольорами  $a$  і  $b$  створює градієнт між ними, тобто новий масив об'єктів-кольорів, у якому перший колір дорівнює  $a$ , останній дорівнює  $b$ , а всі інші є проміжними кольорами.
- метод визначення яскравості кольору, а також метод порівняння кольорів на «більше» – «менше» за значенням яскравості кольору.

Створіть похідний клас, у якому:

- додайте метод для перетворення кольору в модель HSV;
- додайте метод, який у масиві об'єктів-кольорів знаходить той колір, що є найближчим до заданого (за значенням параметра H, «hue»);
- перевизначте метод toString() для виведення кольору у форматі HSV.

4. Напишіть метод для сортування списку кольорів спочатку за величиною червоної компоненти, потім за величиною синьої компоненти. Порядок сортування – за зростанням.

5. Напишіть метод, який у колекції Map видаляє всі повторювані значення (*не ключі*).

### Варіант 4

3. Створіть клас «Користувач соціальної мережі». Внутрішнє представлення користувача: ім'я, логін, пароль, дата народження (java.util.Date), масив постів користувача (де кожен пост – це об'єкт із двома властивостями: дата публікації та текст). Реалізуйте методи класу (статичні) і методи реалізації (нестатичні):

- для визначення віку користувача (цілого числа років) на поточну дату;
- для визначення *рейтингу* користувача як кількості його постів, зроблених за останній місяць;
- метод `toString()`, який видає інформацію користувача у форматі «ім'я – логін».

Створіть похідний клас, у якому:

- додайте статичний метод визначення правильності логіна і пароля (логін має складатися з букв і цифр і починатися з великої літери; пароль має бути не коротшим за 8 символів, обов'язково включати цифри і великі та маленькі літери);
  - перевизначте метод `toString()` для виведення інформації про користувача у форматі «логін – вік – текст останнього за часом посту».
4. Напишіть метод для сортування списку користувачів спочатку за рейтингом, потім за віком (від старших до молодших).
  5. Напишіть метод, який виводить вміст колекції Map (тобто пари ключ–значення) в алфавітному порядку значень, тобто сортує записи за значенням (*не за ключем*).

### **Варіант 5**

3. Створіть клас «Матриця». Внутрішнє представлення – двовимірний масив чисел. Реалізуйте методи класу (статичні) і методи реалізації (нестатичні):
  - для створення матриці заданого розміру та її заповнення числами (з перевіркою на неприпустимі значення);
  - метод знаходження поелементного множення двох матриць;
  - метод `toString()`, який видає інформацію про матрицю у форматі «Матриця  $n*m$ ».

Створіть похідний клас, у якому:

- перевизначте метод «множення» для знаходження матричного множення замість звичайного;
- додайте метод знаходження зворотної матриці;

– перевизначте метод toString() для виведення елементів матриці за рядками (рядки розділяйте символом '\n').

4. Напишіть метод для сортування списку матриць за величиною максимального елемента.

5. Напишіть метод, який у колекції Map видаляє всі повторювані значення (*не ключі*).

### **Варіант 6**

3. Створіть клас «Адреса». Внутрішнє представлення – індекс (5-значне ціле число), місто, вулиця, номер будинку, номер квартири. Реалізуйте методи класу (статичні) і методи реалізації (нестатичні):

– метод створення об'єкта-адреси на основі рядка виду «індекс, вулиця, будинок, квартира місто», де індекс і квартира можуть бути не вказані, наприклад: «34668 Вінниця, вул. Інженерна, 25», «Вінниця, вул. Інженерна, 25, кв. 32»;

– метод, що порівнює дві адреси та визначає, яка частина адреси збігається (повертає спільну частину адреси у вигляді рядка);

– метод перетворення адреси в рядок виду «04123 просп. Перемоги, 35, м. Київ».

Створіть похідний клас, у якому:

– додайте метод, що має параметром масив адрес, у якому адреси можуть повторюватись, і повертає масив адрес без повторень;

– перевизначте метод toString() для виведення адреси англійською мовою. При перекладі на англійську мову російські букви мають транслітеруватись, наприклад, «Chernigiv» замість «Чернігів».

4. Напишіть метод для сортування списку адрес за алфавітом – спочатку за назвою міста, потім за вулицею.

5. Напишіть метод, який у колекції Map підраховує кількість значень (*не ключів*) без урахування повторень.



### **Варіант 7**

3. Створіть клас «Дата». Внутрішнє представлення дати – три цілі числа, що задають день, місяць і рік. Реалізуйте методи класу (статичні) і методи реалізації (нестатичні):

- для створення дати за заданими трьома цілими числами (з перевіркою на неприпустимі значення);
- метод для знаходження дати, наступної за даною (з урахуванням кількості днів у місяці: наприклад, наступною датою за 30 числом місяця може бути 31 або 1 число);
- метод знаходження кількості днів, які розділяють дві задані дати.

Створіть похідний клас, у якому:

- додайте метод визначення дня тижня;
- перевизначте метод `toString()` для виведення дати у форматі «дд.мм.рррр – день\_тижня».

4. Напишіть метод для сортування списку дат спочатку за роком, потім за місяцем.

5. Напишіть метод, який у колекції `Map` підраховує кількість значень (*не ключів*) без урахування повторень.

### **Варіант 8**

3. Створіть клас «Еліпс». Внутрішнє представлення еліпса – дійсні координати вершин описаного навколо нього прямокутника. Реалізуйте методи класу (статичні) і методи реалізації (нестатичні):

- для знаходження координат центру еліпса, площі еліпса;
- для знаходження відстані між центрами двох еліпсів.

Створіть похідний клас «окружність», у якому:

- перевизначте метод `toString()` для виведення окружності у вигляді «Окружність із центром  $(x, y)$  та радіусом  $r$ »;
- додайте метод, що генерує масив точок, що лежать на окружності (кількість точок передається як параметр);

- додайте метод перевірки, чи лежить задана точка всередині окружності.
- 4. Напишіть метод для сортування списку еліпсів за величиною їх площі.
- 5. Напишіть метод, що приймає як параметри дві колекції Map і видаляє з першої колекції все ключі, які зустрічаються у другій колекції.

### **Варіант 9**

- 3. Створіть клас «Час» для зберігання значення часу доби. Внутрішнє представлення часу – три цілих числа, які визначають години, хвилини і секунди. Реалізуйте методи класу (статичні) і методи реалізації (нестатичні):

- для створення об'єкта за заданими трьома цілими числами (з перевіркою на неприпустимі значення);
- метод порівняння об'єктів на рівність, метод визначення кількості секунд, що пройшли з початку доби;
- метод знаходження кількості секунд, які розділяють два задані об'єкти часу.

Створіть похідний клас, у якому:

- додайте метод для визначення відповідного часу в іншому часовому поясі (номер поясу передається як параметр);
- перевизначте метод toString() для виведення часу в текстовому вигляді, наприклад, «десять годин двадцять п'ять хвилин» замість «10:25:00».

- 4. Напишіть метод для сортування списку значень часу спочатку за кількістю годин, потім за кількістю хвилин.
- 5. Напишіть метод, який виводить вміст колекції Map (тобто пари ключ–значення) в алфавітному порядку значень, тобто сортує записи за значенням (*не за ключем*).

### **Варіант 10**

- 3. Створіть клас «Трикутник». Внутрішнє представлення трикутника – цілочисельні координати його вершин. Реалізуйте методи класу (статичні) і методи реалізації (нестатичні) для:

- знаходження площі трикутника, знаходження довжини його сторін;
- перевірки, чи лежить задана точка всередині трикутника.

Створіть похідний клас, у якому:

- перевизначте метод `toString()` для виведення трикутника в вигляді трьох чисел – довжин його сторін;
- додайте метод перевірки, чи перетинається трикутник із іншим заданим трикутником.

4. Напишіть метод для сортування списку трикутників за величиною найдовшої сторони.
5. Напишіть метод, що приймає як параметр колекцію `Map` і будує нову колекцію `Map`, у якій ключі і значення міняються місцями. Всі повторювані значення з вихідної колекції (які не вдалося зберегти в новій колекції) потрібно вивести окремо.

### **Варіант 11**

3. Створіть клас «Ламана лінія». Внутрішнє представлення лінії – список (`ArrayList`) цілочисельних координат її вузлових точок. Метод `toString()` повинен виводити лінію у вигляді «лінія з точки  $(x_1, y_1)$  до точки  $(x_n, y_n)$ ». Реалізуйте методи класу (статичні) і методи реалізації (нестатичні):
  - для додавання і видалення вузлів (у довільному місці лінії);
  - для знаходження загальної довжини лінії;
  - для виведення в рядок послідовності вузлових точок лінії.

Створіть похідний клас, у якому:

- перевизначте метод `toString()` для виведення лінії у вигляді: «лінія довжини  $x$  з точки  $(x_1, y_1)$  до точки  $(x_n, y_n)$ »;
  - додайте метод перевірки, чи не виходить лінія за межі заданого прямокутника.
4. Напишіть метод для сортування списку ліній за кількістю вузлів у ній. Порядок сортування – від коротких ліній до довгих.

5. Напишіть метод, що приймає як параметри дві колекції Map і видаляє з першої колекції все ключі, які зустрічаються у другій колекції.

### **Варіант 12**

3. Створіть клас «Автобусний маршрут». Внутрішнє представлення маршруту – список (List) або масив назв зупинок (пунктів, якими проходить маршрут), а також масив інтервалів за часом між сусідніми зупинками. Реалізуйте методи класу (статичні) і методи реалізації (нестатичні):

- для додавання нових зупинок;
- для виведення маршруту в рядок у вигляді «Початковий\_пункт – пункт1 – Пункт2 ... Кінцевий\_пункт».

Створіть похідний клас, у якому:

- додайте метод для знаходження часу проїзду між будь-якими двома заданими зупинками;
- перевизначте метод toString() для виведення маршруту у вигляді «Маршрут від Початковий\_пункт до Кінцевий\_пункт займає X хвилин».

4. Напишіть метод для сортування списку маршрутів за часом проїзду маршрутом від початку до кінця. Порядок сортування – від коротких маршрутів до довгих.
5. Напишіть метод, що приймає як параметри дві колекції Map і створює нову колекцію Map, до якої потрапляють тільки ті ключі, які присутні в першій колекції і відсутні у другій.

### **Варіант 13**

3. Створіть клас «Користувач соціальної мережі». Внутрішнє представлення користувача: ім'я, логін, пароль, дата народження (java.util.Date), масив постів користувача (де кожен пост – це об'єкт із двома властивостями: дата публікації та текст). Реалізуйте методи класу (статичні) і методи реалізації (нестатичні):

- для визначення віку користувача (цілого числа років) на поточну дату;
- для визначення *рейтингу* користувача як кількості його постів, зроблених за останній місяць;
- метод `toString()`, який видає інформацію користувача у форматі «ім'я – логін».

Створіть похідний клас, у якому:

- додайте статичний метод визначення правильності логіна і пароля (логін має складатися з букв і цифр і починатися з великої літери; пароль має бути не коротшим за 8 символів, обов'язково включати цифри і великі та маленькі літери);
  - перевизначте метод `toString()` для виведення інформації про користувача у форматі «логін – вік – текст останнього за часом посту».
4. Напишіть метод для сортування списку користувачів спочатку за датою останнього посту, потім за ім'ям (за алфавітом).
  5. Напишіть метод, який виводить вміст колекції Map (тобто пари ключ–значення) в алфавітному порядку значень, тобто сортує записи за значенням (*не за ключем*).

### **Варіант 14**

3. Створіть клас «Колір». Внутрішнє представлення кольору – три цілі числа в діапазоні  $[0; 100]$ , що задають відсоток яскравості червоної, зеленої і синьої складових. Реалізуйте методи класу (статичні) і методи реалізації (нестатичні):
  - для створення кольору за заданими трьома цілими числами в діапазоні  $[0; 255]$ ;
  - для створення кольору за трьома дійсними числами в діапазоні  $[0; 100]$  або  $[0; 1]$ ;
  - метод, що за заданими двома кольорами  $a$  і  $b$  створює градієнт між ними, тобто новий масив об'єктів-кольорів, у якому перший колір дорівнює  $a$ , останній дорівнює  $b$ , а всі інші є проміжними кольорами;

- метод визначення яскравості кольору, а також метод порівняння кольорів на «більше» – «менше» за значенням яскравості кольору.

Створіть похідний клас, у якому:

- додайте метод для перетворення кольору в модель HSV;
- додайте метод, який у масиві об'єктів-кольорів знаходить той колір, що є найближчим до заданого (за значенням параметра H, «hue»);
- перевизначте метод toString() для виведення кольору у форматі HSV.

4. Напишіть метод для сортування списку кольорів за величиною яскравості – від темних до світлих.

5. Напишіть метод, що приймає як параметри дві колекції Map і створює нову колекцію Map, до якої потрапляють тільки ті ключі, які присутні в обох вхідних колекціях.

### **Варіант 15**

3. Створіть клас «Адреса». Внутрішнє представлення – індекс (5-значне ціле число), місто, вулиця, номер будинку, номер квартири. Реалізуйте методи класу (статичні) і методи реалізації (нестатичні):

- метод створення об'єкта-адреси на основі рядка виду «індекс, вулиця, будинок, квартира місто», де індекс і квартира можуть бути не вказані, наприклад: «34668 Вінниця, вул. Інженерна, 25», «Вінниця, вул. Інженерна, 25, кв. 32»;
- метод, що порівнює дві адреси та визначає, яка частина адреси збігається (повертає спільну частину адреси у вигляді рядка);
- метод перетворення адреси в рядок виду «04123 просп. Перемоги, 35, м. Київ».

Створіть похідний клас, у якому:

- додайте метод, що приймає як параметри масив адрес, у якому адреси можуть повторюватись, і повертає масив адрес без повторень;

- перевизначте метод `toString()` для виведення адреси на англійській мові. При перекладі на англійську мову російські букви мають транслітеруватись, наприклад, «Chernigiv» замість «Чернігів».
- 4. Напишіть метод для сортування списку адрес за довжиною текстового представлення адреси (`toString()`). Порядок сортування – в напрямку зменшення довжини рядка.
- 5. Напишіть метод, що приймає як параметр колекцію `Map` і будує нову колекцію `Map`, в якій ключі і значення міняються місцями. Всі повторювані значення з вихідної колекції (які не вдалося зберегти в новій колекції) потрібно вивести окремо.

### ***Додаткові завдання***

- 6. Для свого варіанту індивідуального завдання опишіть приклади ситуацій, коли некоректна реалізація методів `Object` для вашого класу може призводити до безповоротної втрати об'єктів цього класу при спробі зберегти їх у колекціях.
- 7. Використовуючи існуючий клас `HashMap`, створіть власну реалізацію інтерфейсу `Map`, яка за будь-якого способу проходу нею буде гарантовано видавати ключі в тому ж порядку, у якому вони додавалися в колекцію.

## Додаток А. колекції JCF

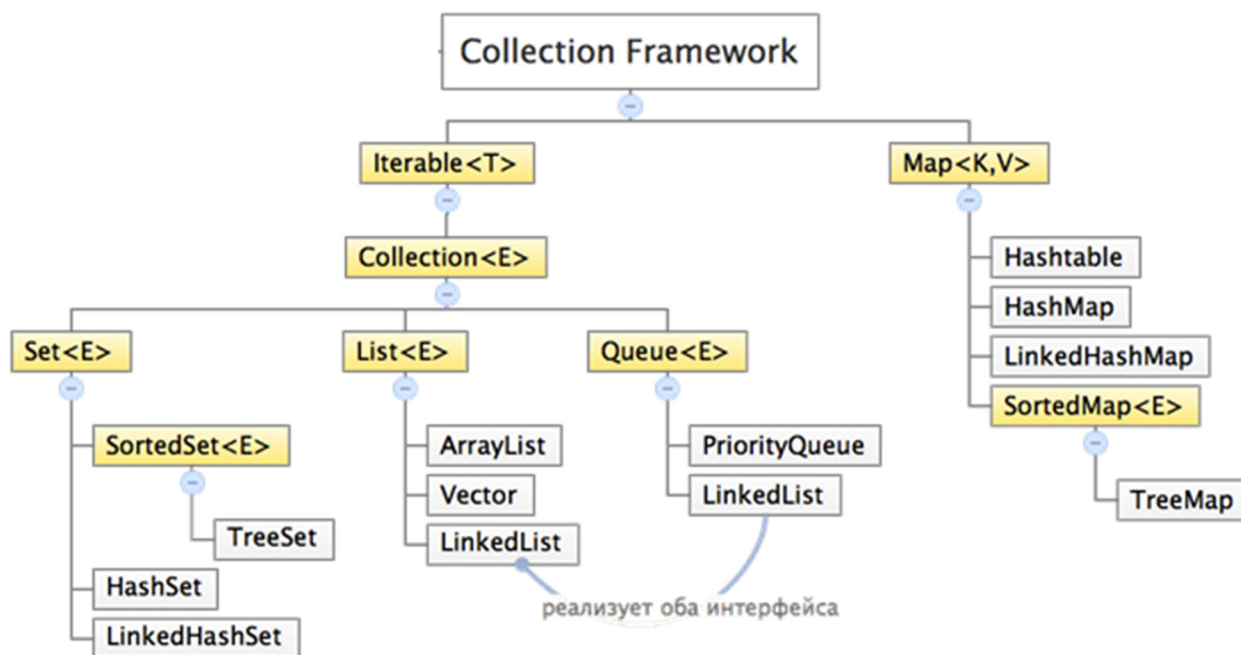


Рис. 3.1 – Ієрархія інтерфейсів JCF і їх реалізацій

## Додаток Б. Виключення

### Загальні класи виключень (java.util)

#### Виключення, що перевіряються (Exception)

ClassNotFoundException	Клас не знайдено (наприклад, бібліотека, в якій він реалізований, недоступна під час виконання програми)
CloneNotSupportedException	Спроба клонувати об'єкт, який не реалізує інтерфейс Cloneable
IllegalAccessException	Заборонено доступ до класу
InstantiationException	Спроба створити об'єкт абстрактного класу або інтерфейсу
InterruptedException	Потік виконання програми перерваний іншим потоком
NoSuchFieldException	Запитуване поле не існує
NoSuchMethodException	Запитуваний метод не існує
ReflectiveOperationException	Виключення, пов'язане з рефлексією



### *Виключення, що не перевіряються (RuntimeException)*

ArithmeticException	Арифметична помилка, наприклад, ділення на нуль
ArrayIndexOutOfBoundsException	Вихід індексу за межі масиву (неприпустимий номер комірки)
ArrayStoreException	Присвоєння елементу масиву об'єкта несумісного типу
ClassCastException	Неправильне приведення типу. Клас об'єкта не відповідає типу, до якого він приводиться
EnumConstantNotPresentException	Спроба використання невизначеного значення перерахування (enum)
IllegalArgumentException	Використання неправильного значення для аргументу при виклику методу
IllegalStateException	Некоректний стан об'єкта, за якого виклик методу неможливий
IllegalThreadStateException	Запитувана операція несумісна з поточним станом потоку
InterruptedException	Сталося переривання потоку виконання в багатопотоковому додатку. Переривання може вивести потік зі стану очікування (wait) або сну (sleep)
IndexOutOfBoundsException	Значення індексу вийшло за допустимі межі. Має два дочірні класи: <code>ArrayIndexOutOfBoundsException</code> і <code>StringIndexOutOfBoundsException</code>
NegativeArraySizeException	Спроба створити масив від'ємного розміру
NullPointerException	Спроба звернутися до об'єкта за посиланням null
NumberFormatException	Помилка при перетворенні рядка в числовий формат
SecurityException	Порушення безпеки
StringIndexOutOfBounds	Вихід індексу за межі рядка (неприпустимий номер символу)

TypeNotPresentException	Клас, запитаний за ім'ям (у вигляді рядка), не знайдено. На відміну від ClassNotFoundException, це виключення не перевіряється
UnsupportedOperationException	Виклик непідтримуваної операції, наприклад, якщо об'єкт підтримує (тобто реалізує) не всі методи загального інтерфейсу

### **Правила коректної обробки виключень**

1. Обробляйте виключення *тільки* в межах відповідальності поточного класу і поточного методу одним із таких способів:
  - 1.1. Обробити на цьому рівні абстракції і заново викинути виключення того ж класу: `catch (Type e) { ... throw e; }`
  - 1.2. Обробити на цьому рівні абстракції і викинути виключення іншого класу: `throw new SomeException("message", e);` – тобто виключення-обгортку вищого рівня;
  - 1.3. Не викидати нових виключень – *тільки* якщо на поточному рівні проблема вирішується повністю.
2. Обробляйте кожен клас виключень окремо.
3. На початку мають знаходитися обробники для більш специфічних класів виключень, потім – для більш загальних.
4. Останній обробник – `catch (Exception e)` – може обробляти *всі* виключення, які не були оброблені раніше. Використовується нечасто, оскільки суперечить правилу 2.
5. Уникайте порожніх обробників ( `catch (Type e) {}` ). Кожне виключення вимагає явних дій.
6. Якщо виключення ігнорується ( `catch (Type e) {}` ), то задокументуйте, чому.
7. Використовуйте блок `finally` тільки для звільнення ресурсів (`close()`, `release()` і т. д.).
8. Обробляйте виключення, що виникають у блоках `catch()` і `finally`.