

Лабораторна робота 4: Хеш-таблиця

Мета роботи:

- 1) дослідити абстрактну структуру даних «Множина» та її реалізацію через хеш-таблицю
- 2) вивчити способи розв'язання колізій у хеш-таблицях через відкриту адресацію і методом ланцюжків
- 3) отримати практичні навички з реалізації хеш-таблиці на мові Java
- 4) навчитися аналізувати реалізації абстрактної структури даних «Множина» та обирати програмну реалізацію відповідно до визначених функціональних вимог та обмежень прикладної задачі.

Інструментальні і програмні засоби:

Java JDK 7 і вище
IDE Eclipse

Склад матеріалів для виконання лабораторної роботи

пакет **common**
 інтерфейс **FigureSet**
 клас **GeomFigure**
пакет **lab4_task1**
 клас **HashTable**
 клас **Task1Main**
 клас **TestTask1**
пакет **lab4_task2**
 клас **SchHashTable**
 клас **OAHashTable**
 клас **Task2Main**

Загальні рекомендації:

Індивідуальні завдання наведені у розділі «Варіанти індивідуальних завдань до лабораторної роботи 4». Робота складається з двох завдань, які виконуються послідовно. В обох завданнях треба описати хеш-таблицю, яка реалізує множину унікальних геометричних фігур **FigureSet**.

У першому та другому завданні обов'язково реалізувати методи, які містять тег **//TODO**. В інтерфейс **Set** з пакету **common** додавати додаткові методи не треба.

Геометричну фігуру описує клас **GeomFigure**. Його поля повинні містити дані відповідно типу фігури згідно з варіантом завдання (таблиця 1).

Клас містить метод `hashCode`, який повертає ціле додатне значення хеш-коду геометричної фігури, який обчислюється на основі всіх полів. В роботі до значення попереднього хеш-коду помножувати на 31 і додавати наступне округлене значення поля. Метод `equals` порівнює геометричні фігури, порівнюючи всі значення всіх полів. У рівних фігур мають бути однакові хеш-коди.

У першому завданні хеш-таблиця не розв'язує колізії, які виникають при додаванні елементів, в яких однакові хеш-значення. Іншими словами, якщо комірка зайнята, то елемент у таблицю не додається. При пошуку і вилученні елементів перевіряється тільки одна комірка для хеш-значення об'єкта.

Хеш-значення об'єкта (геометричної фігури) обчислюється хеш-функцією у методі `hash` класу, який описує хеш-таблицю. У цьому методі спочатку береться хеш-код об'єкта методом `hashCode` класу `GeomFigure`, а потім це значення використовується при обчисленнях за методом ділення або множення згідно з варіантом завдання.

У другому завданні хеш-таблиця не розв'язує колізію за методом, визначеному у варіанті індивідуального завдання (таблиця 2). Перед додаванням елемента важливо перевірити, чи є вже такий елемент у множині. Якщо є, то новий елемент не додається.

Колізії оброблюються при додаванні, пошуку та видаленні елемента, якщо елементи мають однакові хеш-значення. При додаванні елемента і після його перевірки на унікальність перевіряється, чи зайнята комірка, яка відповідає хеш-значенню об'єкта. Порожня – елемент у таблицю додається одразу, але якщо зайнята, то об'єкт в таблицю все ж таки додається, але вільна комірка шукається за певним правилом відповідно до методу розв'язання колізій: відкрита адресація або метод ланцюжків (таблиця 2). При пошуку і вилученні елементів з таблиці спочатку перевіряється комірка, яка відповідає хеш-значенню об'єкта. Якщо там не міститься шуканий елемент, то пошук елемента продовжується за тим же правилом, що і при додаванні елемента.

Для перевірки на рівність об'єктів геометричних фігур в операціях пошуку та вилучення елементів хеш-таблиці необхідно викликати метод `equals`. Наприклад, порівняти фігури `gf1` і `gf2` можна `gf1.equals(gf2)` або `gf2.equals(gf1)`. Обидва виклики мають повертати однаковий результат: `true`, якщо об'єкти рівні, або `false`, якщо ні.

Працездатність програми необхідно перевірити у клієнтських класах `Task1Main` і `Task2Main`. Для цього треба прочитати дані з файлу `csv`, створити на основі цих даних таблицю і вивести її вміст. В файл треба записати різноманітні дані, включаючи унікальні, не унікальні фігури, а також ті, що мають однакові хеш-значення. При виведенні хеш-таблиці треба звернути увагу, як вона буде форматована: на одному рядку треба вивести номер комірки таблиці, хеш-код об'єкта, та його дані. Якщо комірка порожня, то напроти номера комірки треба вивести повідомлення, що вона порожня.

До того ж, для тестування методів класу хеш-таблиці треба написати модульні тести. Каркас модульних тестів надано у файлах **TestTask1** і **TestTask1**. Для того, щоб провести модульне тестування, до проекту треба підключити бібліотеку JUnit: з контекстного меню проекту обороти **Build Path -> Add Library -> JUnit**.

Опис інтерфейсів і класів пакету common:

Інтерфейс FigureSet специфікує методи для обробки множини геометричних фігур:

boolean add(GeomFigure gf) – додає фігуру **gf** у множину, якщо такої фігури у множині немає, і повертає **true**. Якщо множина не змінилася, повертає **false**

boolean contains(GeomFigure gf) – перевіряє, чи є фігура **gf** у множині. Якщо є, повертає **true**, в іншому випадку повертає **false**

boolean remove(GeomFigure gf) – вилучає фігуру **gf** з множини, якщо вона є у множині, і повертає **true**. Якщо множина не змінилася, повертає **false**

int size() – повертає кількість елементів у множині

boolean isEmpty() – повертає **true**, якщо у множині немає елементів

Клас GeomFigure представляє опис інформації про геометричну фігуру і містить *поля*:

value – деяке значення, що описує фігуру. При виконанні лабораторної роботи треба описати дані відповідно до варіанту;

hash – хеш-код об'єкта;

методи:

public int hashCode() – обчислює хеш-код об'єкта

public String toString() – повертає рядок, який містить всі дані об'єкта у форматovanому вигляді

public boolean equals(GeomFigure gf) – порівнює даний об'єкт з тим, що передано як параметр

Опис класу наведено як приклад. Для виконання роботи початковий код класу треба переписати, виходячи з варіанту індивідуального завдання (табл.1, стовпчик 2).

Завдання

Завдання 4.1. Реалізувати хеш-таблицю, яка зберігає об'єкти типу **GeomFigure** і не оброблює колізії в операціях додавання, пошуку та вилучення елементів

Порядок виконання

- 1) Описати методи класу **GeomFigure** згідно з варіантом завдання
- 2) Реалізувати методи класу **Hastable**
- 3) Перевірити працездатність програми у клієнтському класі **Task1Main**:
 - створити файл з даними про геометричні фігури
 - зчитати дані з файлу і створити масив фігур

- створити таблицю і заповнити її зчитаними з файлу даними
 - вивести вміст таблиці
- 4) Тестувати операції додавання, пошуку та вилучення елементів з хеш-таблиці за допомогою модульних тестів у файлі **TestTask1**. Звернути увагу на такі тестові випадки:
- додавання елемента у порожню хеш-таблицю
 - додавання елементів у хеш-таблицю зі збільшенням її розміру
 - додавання унікальних і не унікальних елементів
 - додавання елементів, які мають однакові хеш-значення
 - вилучення елемента з порожньої хеш-таблиці
 - вилучення, які є та відсутні у хеш-таблиці
 - вилучення елементів, які мають однакові хеш-значення
 - пошук елемента у порожній хеш-таблиці
 - пошук елементів, які є та відсутні у хеш-таблиці
 - пошук елементів, які мають однакові хеш-значення

Завдання 4.2. Реалізувати хеш-таблицю, яка зберігає об'єкти типу **GeomFigure** і обробляє колізії в операціях додавання, пошуку та вилучення елементів за методом згідно з варіантом (табл.2)

якщо колізії розв'язуються методом ланцюжків

- 1) Описати клас **DLNode** для опису вузла таблиці
- 2) Реалізувати методи класу **SchHastable**

якщо колізії розв'язуються методом відкритої адресації

- 1) Реалізувати методи класу **OAHastable**
- 3) Перевірити працездатність програми у клієнтському класі **Task2Main**. Необхідні методи можна імпортувати з пакету **lab4_task1**. Тестові файли використовувати ті самі, що і в першому завданні. Порядок дій такий, як і в **Task1Main**.
- 4) Створити файл **TestTask2** і написати тести для операцій додавання, пошуку та вилучення елементів з хеш-таблицю
- 5) Порівняти результати тестування операцій залежно від того, у якому порядку елементи додавалися у хеш-таблицю і вилучалися з хеш-таблиці

Питання для самоперевірки:

- 1) Що таке хеш-таблиця? За якими критеріями можна класифікувати хеш-таблиці?
- 2) Для чого призначено хешування? Які є методи хешування?
- 3) За яких умов у хеш-таблиці виникають колізії? Якими способами вони вирішуються?
- 4) Як реалізовано хеш-таблицю з відкритою адресацією в разі лінійного зондування?
- 5) Як реалізовано хеш-таблицю з відкритою адресацією в разі подвійного зондування?
- 6) Як реалізовано хеш-таблицю з відкритою адресацією в разі квадратичного зондування?
- 7) Як реалізовано хеш-таблицю з роздільним зв'язуванням?
- 8) Як і за яких умов виконується операція повторного хешування?

Таблиця 1. Варіанти індивідуальних завдань лабораторної роботи 4

Варіант	Геометрична фігура	Метод хешування
1	2	3
1	Трикутник: координати вершин, площа, периметр	Ділення
2	Вектор: координати кінця вектора, полярні координати	Множення
3	Прямокутник: координати вершин, площа, периметр	Ділення
4	Відрізок: координати початку і кінця, довжина і кут з віссю OX	Множення
5	Квадрат: координати вершин, площа, периметр	Ділення
6	Трапеція: координати вершин, площа, периметр	Множення
7	Ромб: координати вершин, площа, периметр	Ділення
8	Коло: координати центру і радіус, площа, периметр	Множення
9	Паралелограм: координати вершин, площа, периметр	Ділення
10	Трикутник: координати вершин, площа, периметр	Множення
11	Трапеція: координати вершин, площа, периметр	Ділення
12	Квадрат: координати вершин, площа, периметр	Множення
13	Вектор: полярні координати, координати кінця вектора	Ділення
14	Коло: координати центру і радіус, довжина кола	Множення
15	П'ятикутник: координати вершин, периметр	Ділення
16	Прямокутник: координати вершин, площа, периметр а	Множення
17	Відрізок: координати початку і кінця, довжини і кут з віссю OX	Ділення
18	Ромб: координати вершин, площа, периметр	Множення
19	Вектор: координати кінця вектора, полярні координати	Ділення
20	Куля: радіус, площа поверхні сфери, яка обмежує кулю, об'єм	Множення
21	Куб: довжина ребра, площа поверхні та об'єм	Ділення
22	Циліндр: висота, радіус основи, загальна площі та об'єм	Множення
23	Сектор: радіус, кут, площа	Ділення
24	Прямокутний паралелепіпед: виміри паралелепіпеда, об'єм, площа бічної і повної поверхні	Множення

Таблиця 2. Варіанти індивідуальних завдань лабораторної роботи 4 (частина 2)

Варіант	Метод розв'язання колізій
1, 5, 9, 13, 16, 20	Роздільне зв'язування
2, 6, 10, 17, 21, 24	Відкрита адресація – лінійне зондування
3, 7, 11, 15, 19, 22	Відкрита адресація – подвійне хешування
4, 8, 12, 14, 18, 23	Відкрита адресація – квадратичне зондування