



Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

ЛАБОРАТОРНА РОБОТА № 1
з дисципліни «Мова програмування Java»
Тема: «Робота з циклами, масивами та рядками в Java»
Варіант 119

Виконала:

Студентка групи ІА-31

Соколова Поліна

Мета роботи – здобути досвід роботи з Java Development Kit та обраним інтегрованим середовищем розробки при практичному вирішенні обраного завдання, ознайомитись з процесом виконання Java програм.

Завдання:

9. Знайти всі досконалі числа в діапазоні від 1 до деякого заданого позитивного числа (досконале число – це число, яке дорівнює сумі всіх своїх дільників, крім самого себе. Наприклад, $6 = 1+2+3$).

Клас `PerfectNumberFinder` містить всю логіку для роботи з досконалими числами.

У методі `isPerfect()` перевіряємо на валідність: числа ≤ 1 не можуть бути досконалими, перебираємо дільники тільки до `number / 2`, оскільки дільники більші за половину числа (крім самого числа) не існують, перевіряємо чи число ділиться без залишку. Змінна `sum` зберігає суму всіх знайдених дільників, порівнюємо суму з вихідним числом, повертаємо `true` якщо число досконале, `false` якщо ні.

У методі `findPerfectNumbers()` використовуємо колекцію `ArrayList` для зберігання результатів, цикл `for` перебирає всі числа від 1 до `maxNumber`, викликаємо метод `isPerfect()` для перевірки чисел, додаємо до списку, якщо досконале число.

Метод `getDivisors()` для отримання всіх дільників числа.

Клас `Main` забезпечує взаємодію з користувачем через консоль.

Клас `Scanner` використовується для читання вводу користувача, виконуємо валідацію введення для перевірки чи є введене число додатним.

Створюємо список досконалих чисел і виводимо результат.

Також реалізовано unit-тести для перевірки коректності роботи програми.

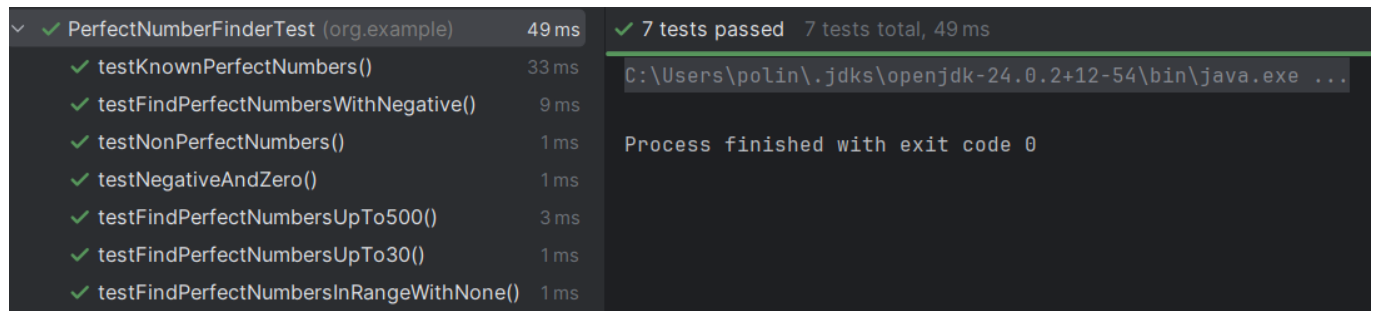
Для цього є клас `PerfectNumberFinderTest`. У ньому будемо використовувати анотацію `@Test`, що позначає метод як тестовий,

метод `assertTrue()`: перевіряє що результат дорівнює `true`,

метод `assertFalse()`: перевіряє що результат дорівнює `false`,

метод `assertEquals()`: порівнює два значення на рівність.

Тести пройшли успішно:



The screenshot shows the test results for the `PerfectNumberFinderTest` class. On the left, a list of seven tests is shown, all of which passed. The tests and their durations are: `testKnownPerfectNumbers()` (33 ms), `testFindPerfectNumbersWithNegative()` (9 ms), `testNonPerfectNumbers()` (1 ms), `testNegativeAndZero()` (1 ms), `testFindPerfectNumbersUpTo500()` (3 ms), `testFindPerfectNumbersUpTo30()` (1 ms), and `testFindPerfectNumbersInRangeWithNone()` (1 ms). On the right, a summary indicates that 7 tests passed out of 7 tests total, taking 49 ms. Below this, the command prompt path `C:\Users\polin\.jdk\openjdk-24.0.2+12-54\bin\java.exe` is visible, followed by the message "Process finished with exit code 0".

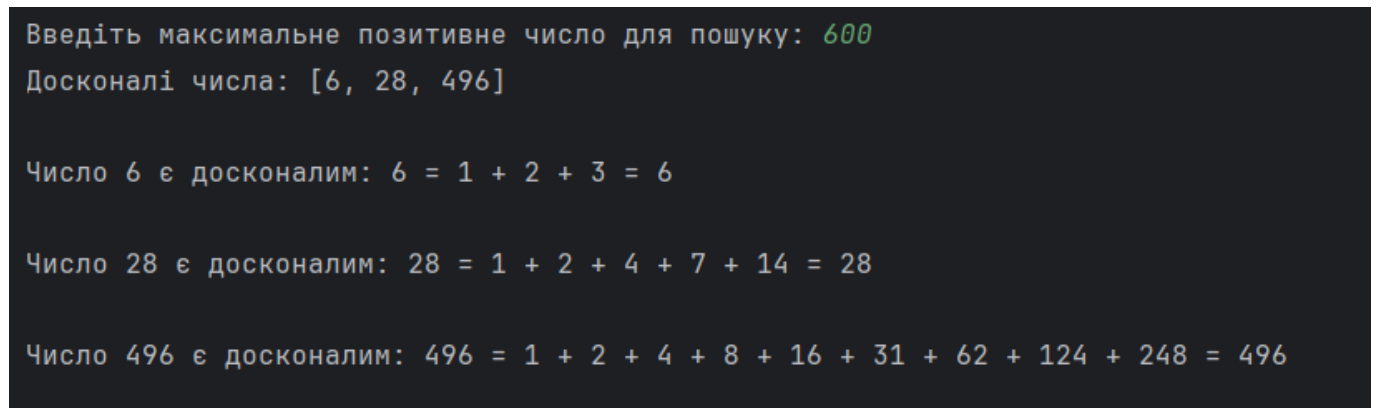
Test Name	Duration
✓ <code>testKnownPerfectNumbers()</code>	33 ms
✓ <code>testFindPerfectNumbersWithNegative()</code>	9 ms
✓ <code>testNonPerfectNumbers()</code>	1 ms
✓ <code>testNegativeAndZero()</code>	1 ms
✓ <code>testFindPerfectNumbersUpTo500()</code>	3 ms
✓ <code>testFindPerfectNumbersUpTo30()</code>	1 ms
✓ <code>testFindPerfectNumbersInRangeWithNone()</code>	1 ms

✓ 7 tests passed 7 tests total, 49 ms

C:\Users\polin\.jdk\openjdk-24.0.2+12-54\bin\java.exe ...

Process finished with exit code 0

Результат роботи:



The screenshot shows the output of a Java program. It prompts the user to enter a maximum positive number for search, with `600` entered. The program then lists the perfect numbers found: `[6, 28, 496]`. For each perfect number, it displays the equation showing the sum of its proper divisors equals the number itself: `6 = 1 + 2 + 3 = 6`, `28 = 1 + 2 + 4 + 7 + 14 = 28`, and `496 = 1 + 2 + 4 + 8 + 16 + 31 + 62 + 124 + 248 = 496`.

```
Введіть максимальне позитивне число для пошуку: 600
Досконалі числа: [6, 28, 496]

Число 6 є досконалим: 6 = 1 + 2 + 3 = 6

Число 28 є досконалим: 28 = 1 + 2 + 4 + 7 + 14 = 28

Число 496 є досконалим: 496 = 1 + 2 + 4 + 8 + 16 + 31 + 62 + 124 + 248 = 496
```

Висновок: у результаті виконання лабораторної роботи я освоїла роботу з циклами в Java, навчилася працювати з масивами: використала `ArrayList<Integer>` для динамічного зберігання досконалих чисел та їх дільників, освоїла основні операції з списками (`add()`, `get()`, `size()`, `contains()`). Крім цього реалізувала unit-тести, які перевіряють коректність роботи програми в різних сценаріях.