

Laboratorium 10

Zaimplementuj w języku Java rozwiązania poniższych problemów:

Liczby doskonałe (J1)

8. Znajdź wszystkie liczby doskonałe z przedziału $\langle m;n \rangle$. Oprzyj rozwiązanie bieżącego zadania na dwóch typach klasowych: bazowym `Doskonała` i pochodnym `Liczby_doskonale`.
Dla każdej znalezionej liczby przedstaw na standardowym wyjściu dowód jej doskonałości, w postaci (dla przykładowej liczby doskonałej 28):
 $28 = 1 + 2 + 4 + 7 + 14$.
Dowód ten przeprowadź przy użyciu metody `Dowod` zdefiniowanej w klasie bazowej `Doskonała`.
Utwórz obiekt klasy pochodnej `Liczby_doskonale`. Podstaw ten obiekt za zmienną `ldosk` typu klasowego `Doskonała` (czy taka operacja jest dozwolona?). Podaj dowody doskonałości dla wszystkich znalezionej liczb doskonałych z zakresu $\langle m;n \rangle$.

Liczby doskonałe (J2)

9. Znajdź wszystkie liczby doskonałe z przedziału $\langle m;n \rangle$. Dla każdej znalezionej liczby przedstaw dowód jej doskonałości, w postaci (dla przykładowej liczby doskonałej 28):
2. liczba w przedziale: $28 = 1 + 2 + 4 + 7 + 14$
gdzie początkowa 2 oznacza numer porządkowy liczby doskonałej 28 we wskazanym przedziale $\langle 5; 100 \rangle$.
Zmodyfikuj w tym celu rozwiązanie zadania 10.7 tak, by skorzystać w nim z paradygmatu polimorfizmu. I tak, dotychczasową metodę `Dowod` z klasy bazowej `Doskonała` zdefiniuj jako metodę wirtualną, a następnie dołącz do założonej przez nią rodziny metod wirtualnych – metodę `Dowod` w klasie pochodnej `Liczby_doskonale`.
Utwórz obiekt klasy pochodnej `Liczby_doskonale`. Podstaw ten obiekt za zmienną `ldosk` typu klasowego `Doskonała` (czy taka operacja jest dozwolona? co uzyskujesz?). Podaj dowody doskonałości dla wszystkich znalezionych liczb doskonałych z zakresu $\langle m;n \rangle$. Podaj je:
a) w formie skróconej (bez numeru porządkowego liczby w przedziale),
b) w formie rozszerzonej (z numerem porządkowym liczby w przedziale).

Przetwarzanie tekstów (trudne)

10. Na standardowym wejściu, w dwóch kolejnych wierszach zapisano:
– tzw. pomocniczy łańcuch znaków,
– ścieżkę dostępu do pliku tekstowego, w którym są przechowywane tzw. główne łańcuchy znaków (każdy w odrębnym wierszu, nie więcej niż 50).
Sprawdź, ile razy pomocniczy łańcuch znaków mieści się w każdym z łańcuchów głównych (dwa różne wystąpienia łańcucha pomocniczego nie mogą się pokrywać na żadnym ze znaków).
Zadanie wykonaj przy użyciu typu klasowego `Teksty` zawierającego, między innymi:
– pole prywatne `pomoc`, przechowujące pomocniczy łańcuch znaków,
– metodę prywatną `ile_razy`, sprawdzającą liczbę różnych wystąpień łańcucha pomocniczego w łańcuchu głównym,

- pole publiczne **zestawienie**, przechowujące zbiorczą informację o liczbie wystąpień łańcucha pomocniczego we wszystkich łańcuchach głównych,
- pole publiczne **najlepszy**, przechowujące łańcuch główny o największej wykrytej liczbie wystąpień łańcucha pomocniczego,
- 2-argumentowy konstruktor, z argumentami: pomocniczym łańcuchem znaków i ścieżką dostępu do pliku wejściowego,
- metodę publiczną **czytaj**, odpowiedzialną za wczytanie wszystkich umieszczonych w pliku wejściowym łańcuchów głównych i zrealizowanie żądanych obliczeń.

Wyprowadź na wyjście informację o liczbie wystąpień ciągu pomocniczego w poszczególnych ciągach głównych, numerowanych kolejno od 1, oraz najlepszy ciąg główny.

Utwórz obiekt klasy `Teksty` i zastosuj odpowiednie metody w celu rozwiązania zadania.

Przykład

Dane wejściowe

abca

dane.txt

Zawartość pliku dane.txt

abcabca

nsabcatdftabca

abccabc

abcaabcaewrxrabcayyabc

Dane wyjściowe

1 - 1

2 - 2

3 - 0

4 - 3

abcaabcaewrxrabcayyabc

Wskazówka: Materiały dotyczące konfiguracji środowiska do programowania w Javie, a także samego programowanie w Javie znajdziesz, między innymi, pod adresem:

<https://javastart.pl/baza-wiedzy/>