

Java

(studia wieczorowe)

zadanie nr 2
19 października 2005

Zdefiniuj w jednym pliku klasę `Pascal`, która będzie przechowywać stabilizowane wartości *trójkąta Pascala*, a w drugim pliku klasę `Wzor` z metodą `main()`, która korzystając z trójkąta Pascala będzie wyliczała wzór dwumianowy dla zadanego parametru n .

Trójkąt Pascala powinien być zapamiętany w prywatnej, ostatecznej (finalnej) i statycznej tablicy `trojkat` (macierz trójkątna). Inicjalizacja tej tablicy powinna odbywać się w statycznym bloku inicjalizującym. Dostęp do elementów tablicy powinien być realizowany przez publiczną metodę `po`, która mając dwa parametry n i k , odczytuje k -tą pozycję z n -tego wiersza, czyli wartość $\binom{n}{k}$. Rozmiar tablicy powinien wynosić 34 (największy z możliwych, w którym wszystkie wartości nie przekraczają zakresu typu `int`).

```
public class Pascal
{
    private static final int[][] trojkat = new int[34][34];
    static
    {
        // inicjalizacja tablicy
    }

    public static int po (int n, int k)
    {
        // n po k
    }
}
```

Program napisany w klasie `Wzor` ma za zadanie wypisać wzór dwumianowy dla wyrażenia postaci $(a+b)^n$ lub $(a-b)^n$. Wartości a , b i n mają zostać przekazane do programu przez parametry wywołania (a i b to łańcuchy znaków złożone tylko z liter, a n jest niedużą liczbą naturalną). Oto kilka przykładowych wywołań tego programu:

```
> java Wzor (a-b)^3
a^3 - 3*a^2*b + 3*a*b^2 - b^3
> java Wzor (x+y)^1
x + y
> java Wzor (ve-wu)^0
1
> java Wzor (iks+ygrek)^2
iks^2 + 2*iks*ygrek + ygrek^2
> java Wzor (s-t)^4
s^4 - 4*s^3*t + 6*s^2*t^2 - 4*s*t^3 + t^4
```

Uwaga:

Kolejne wartości w trójkącie Pascala można obliczyć według następującego schematu ($0 \leq k \leq n$):

$$\binom{n}{k} = \begin{cases} 1 & : k = 0 \vee k = n \\ \binom{n-1}{k-1} + \binom{n-1}{k} & : 0 < k < n \end{cases}$$

Uwaga:

W klasie `String` jest zdefiniowanych kilka praktycznych metod, które możesz wykorzystać w tym programie.

Metoda `length()` zwraca długość napisu. Metoda `trim()` obcina białe znaki na początku i na końcu napisu (zwraca nowy napis). Metoda `charAt(int n)` podaje n -ty znak w napisie. Metoda `indexOf(char zn)` podaje pozycję pierwszego wystąpienia znaku zn w napisie (lub wartość -1 gdy znak w ogóle w napisie nie występuje). Metoda `substring(int pocz, int kon)` wycina z napisu podnapis poczynając od pozycji $pocz$ a kończąc na pozycji $kon-1$ (zwraca nowy napis). I wreszcie metoda `matches(String wzor)` sprawdza, czy napis dopasowuje się do podanego wyrażenia regularnego $wzor$ (wyrażeniem regularnym, które reprezentuje dowolny identyfikator złożony z liter lub cyfr i zaczynający się od litery jest `"\p{Alpha}\p{Alnum}*"`).

Paweł Rzechonek
Wrocław, 12 października 2005.