Wstęp do programowania obiektowego - lista 2

Agnieszka Kazimierska, Karol Kulinowski, Marta Hendler

Programy przeznaczone do oceny należy umieszczać w repozytorium kursu w portalu eportal.pwr.edu.pl. Programy należy oddawać w formie plików źródłowych (.java), bez pakowania. Podczas rozwiązywania zadań:

- Zwróć uwagę na typ zmiennych podawanych przez użytkownika ze standardowego wejścia (liczby całkowite/zmiennoprzecinkowe, łańcuchy znaków).
- Zwróć uwagę na dopuszczalne wartości zmiennych podawanych przez użytkownika. Jeżeli podana została zmienna o nieprawidłowej wartości, program powinien wyświetlać odpowiedni komunikat. Jeżeli treść zadania nie określa *explicite* zakresu dopuszczalnych wartości, zastanów się, czy warunki nie wynikają z treści zadania, np. czy dopuszczalne są liczby ujemne lub 0? A puste łańcuchy znaków?
- Zwróć uwagę na typ zmiennych będących wynikiem obliczeń (szczególnie w przypadku dzielenia).

Zadanie 1

Zgodnie z regułą komplementarności zasady w podwójnej nici DNA łączą się w określone pary: guanina (G) z cytozyną (C), a adenina (A) z tyminą (T). Napisz program do wyznaczania komplementarnej nici DNA w oparciu o tę regułę. Program powinien przyjmować łańcuch znaków reprezentujący pierwszą nić, a następnie wyznaczać nić komplementarną i wyświetlać obie nici na standardowym wyjściu. W przypadku, gdy w pierwszej nici podano niedozwolone znaki (tj. inne niż G, C, A i T), program powinien w ich miejsce wstawiać znak błędu (np. X, kropkę, podkreślnik). Napisz program w taki sposób, aby traktował małe i wielkie litery w nici wejściowej jako takie same, a wynik końcowy wyświetlał wielkimi literami.

Zadanie 2

Napisz program do konwersji oznaczeń kolorów z zapisu szestnastkowego (HEX) do modelu RGB. Program powinien przyjmować kod HEX w postaci łańcucha znaków `#RRGGBB`, gdzie `RR`, `GG`, `BB` oznaczają liczby dwucyfrowe w systemie szestnastkowym odpowiadające zawartości koloru czerwonego, zielonego i niebieskiego. Następnie program powinien konwertować każdą z tych liczb do zapisu w systemie dziesiętnym i wyświetlać oznaczenie koloru w postaci `(r, g, b)`. Zwróć uwagę na sprawdzanie poprawności kodu HEX podawanego przez użytkownika, np. czy kod ma odpowiednią długość i zawiera początkowy znak `#`.

Przykład: Dla koloru żółtego po podaniu kodu `#FFFF00` powinien zostać wypisany wynik w postaci `(255, 255, 00)`.

Zadanie 3

Napisz program sprawdzający poprawność nazwy użytkownika zgodnie z następującymi kryteriami: nazwa musi być dłuższa niż 3 znaki, ale krótsza niż 16; nazwa może zawierać małe i wielkie litery oraz cyfry, ale nie można zawierać znaków specjalnych; nazwa nie może zaczynać się od cyfry. Program powinien przyjmować łańcuch znaków reprezentujący proponowaną nazwę użytkownika, a następnie wyświetlać informację o tym, czy nazwa jest poprawna. Jeżeli nazwa nie jest poprawna, program powinien wyświetlać informację, który warunek nie jest spełniony.

Zadanie 4

Napisz program do normalizacji ciągu liczb. Program powinien przyjmować od użytkownika rozmiar tablicy do przechowywania ciągu wejściowego (liczba całkowita) oraz jego elementy (liczby zmiennoprzecinkowe). Następnie program powinien skalować elementy ciagu przechowywane w tablicy jednowymiarowej do zakresu 0–1 zgodnie ze wzorem:

$$y = (x - min)/(max - min), \tag{1}$$

gdzie y to wartość przeskalowana, x - wartość początkowa, a min, max to najmniejsza i największa wartość w podanym ciągu liczb. Program powinien wyświetlać na standardowym wyjściu wynik skalowania bez zmiany kolejności liczb w ciągu.

Zadanie 5

Napisz program generujący macierz (tablicę dwuwymiarową) o rozmiarze 5 x 5 elementów wypełnioną pseudolosowymi liczbami całkowitymi z zakresu od 0 do 100 (włącznie), a następnie obliczający sumę elementów leżących na przekątnej (od lewego górnego do prawego dolnego rogu) oraz przeciwprzekątnej (od prawego górnego do lewego dolnego rogu) tej macierzy. Program powinien wyświetlać na standardowym wyjściu wygenerowaną macierz oraz obliczone sumy elementów.

Zadanie 6

Gra w Sudoku polega na wypełnieniu planszy o rozmiarze 9 x 9 elementów w taki sposób, aby w każdym rzędzie, kolumnie i wewnętrznych tablicach 3 x 3 każda z liczb z zakresu od 1 do 9 pojawiała się dokładnie raz. Napisz program, który sprawdzi, czy wypełniona plansza jest rozwiązana poprawnie i wyświetli wynik sprawdzania na standardowym wyjściu. Rozwiązanie do sprawdzenia może być wpisane "na sztywno"w kodzie lub podawane przez użytkownika.