Zadania z programowania w języku Java dla II roku Informatyki.

dr Agnieszka Zbrzezny

Napisy

1. Do każdego z poniższych punktów napisz odpowiednią metodę oraz odpowiedni program testujący.

   (a) Napisz statyczną metodę `int countChar(String str, char c)`, która zliczy ilość wystąpień znaku `c` w napisie `str`. W funkcji `main` wczytaj napis oraz znak, a następnie wypisz na ekran odpowiedni komunikat.

   (b) Napisz statyczną metodę `int countSubStr(String str, String subStr)`, która zliczy ilość wystąpień napisu `subStr` w napisie `str`. W funkcji `main` wczytaj dwa napisy, a następnie wypisz na ekran odpowiedni komunikat.

   (c) Napisz statyczną metodę `String middle(String str)`, która zwraca napis zawierający środkowy znak w `str`, jeśli długość napisu `str` jest nieparzysta, lub dwa znaki środkowe, jeśli długość napisu `str` jest parzysta. Przykładowo, `middle("middle")` zwraca napis `"dd"`. W funkcji `main` wypisz wynik działania funkcji `middle` dla napisu wprowadzonego z klawiatury.

   (d) Napisz statyczną metodę `String repeat(String str, int n)`, która zwraca napis będący konkatenacją `n` kopii napisu `str`. Przykładowo, `repeat("ho")` zwraca `"hohoho"`. W funkcji `main` wypisz wynik działania funkcji `repeat` dla napisu wprowadzonego z klawiatury.

   (e) Napisz statyczną metodę `int[] where(String str, String subStr)`, która utworzy tablicę indeksów wskazujących kolejne wystąpienia napisu `subStr` w napisie `str`. W funkcji `main` wczytaj dwa napisy, a następnie wypisz na ekran zawartość utworzonej tablicy.

   (f) Napisz statyczną metodę `String change(String str)`, która utworzy zmodyfikowaną wersję napisu `str` poprzez zamianę wszystkich małych liter napisu `str` na odpowiednie duże litery oraz wszystkich dużych liter napisu `str` na odpowiednie małe litery. W metodzie `change` utwórz najpierw obiekt klasy `StringBuffer` (wykorzystaj metodę `append` z klasy `StringBuffer`), a następnie zwróć napis utworzony poprzez użycie metody `toString` z klasy `StringBuffer`. W funkcji `main` wczytaj napis, a następnie wypisz na ekran jego zmodyfikowaną wersję.

   (g) Napisz statyczną metodę `String nice(String str)`, która utworzy zmodyfikowaną wersję napisu `str` reprezentującego liczby całkowite w zapisie dziesiętnym poprzez wstawienie znaku apostrofa co trzy cyfry począwszy od prawej strony. W metodzie `nice` utwórz najpierw obiekt klasy `StringBuffer` (wykorzystaj metodę `append` z klasy `StringBuffer`), a następnie zwróć napis utworzony poprzez użycie metody `toString` z klasy `StringBuffer`. W funkcji `main` wczytaj napis, a następnie wypisz na ekran jego zmodyfikowaną wersję.

   (h) Napisz statyczną metodę `nice` będącą modyfikacji metody `nice` z poprzedniego zadania tak, aby znak separatora oraz liczba pozycji pomiędzy kolejnymi wystąpieniami separatora były przekazywane jako argumenty wywołania.

2. Napisz program, który zliczy ilość wystąpień podanego znaku w podanym pliku Nazwę pliku oraz znak przekaż jako argumenty wywołania programu.

3. Napisz program, który zliczy ilość wystąpień podanego wyrazu w podanym pliku (na potrzeby niniejszego zadania *wyrazem* nazwiemy dowolny ciąg znaków nie zawierający spacji). Nazwę pliku oraz wyraz przekaż jako argumenty wywołania programu.

Klasy `BigInteger` oraz `BigDecimal`

4. Na pierwszym polu szachownicy o rozmiarze `n` $\times$ `n` kładziemy jedno ziarnko maku, na drugim dwa ziarnka maku, na trzecim dwa razy więcej niż na drugim, na czwartym dwa razy więcej niż na trzecim itd. Napisz program, który obliczy ile ziarenek maku położymy w sumie na szachownicy. Liczbę `n` przekaż jako argument wywołania programu. Obliczenia wykonaj stosując obiekty klasy `java.math.BigInteger`.

5. Napisz program, który wyświetli wielkość kapitału po `n` latach, przy założeniu, że odsetki są kapitalizowane po każdym roku, a stopa procentowa nie ulega zmianie w czasie zadeklarowanego okresu oszczędzania. Kapitał początkowy `k`, stopę procentową `p` oraz długość okresu oszczędzania w latach `n` przekaż jako argumenty wywołania programu. Obliczenia z dokładnością do dwóch miejsc po przecinku wykonaj stosując obiekty klasy `java.math.BigDecimal`.