

Programowanie zaawansowane, lab. 5

1. Zdefiniuj klasę przetwarzającą numer PESEL. Ma ona w konstruktorze przyjmować numer PESEL w postaci obiektu klasy `String`. Powinna ona mieć metody pozwalające na wyciągnięcie z numeru pełnej daty urodzenia, roku, miesiąca, dnia, płci oraz sprawdzającą, czy numer jest poprawny (powinien sprawdzać długość, poprawność miesiąca, dnia urodzenia oraz sumę kontrolną).
2. Utwórz klasę statyczną **`StringParser`**, która będzie w sobie zawierała statyczne metody do przetwarzania napisów (zatem nie trzeba tworzyć obiektu tej klasy, żeby ich używać).

- Napisz funkcję **`static int sumalter(String s)`** obliczającą sumę cyfr liczby całkowitej podanej jako argument `s` w sposób iteracyjny.
- Napisz funkcję **`static int sumaRek(String s)`** obliczającą sumę cyfr liczby całkowitej podanej jako argument `s` w sposób rekurencyjny.
- Napisz funkcję **`static String sumaBinarna(String s1, String s2)`** obliczającą sumę liczb dwójkowych podanych jako argumenty `s1` i `s2` oraz zwracającą wynik jako nowy binarny napis.

Wskazówka: możesz wykorzystać konstrukcje `Integer.parseInt(dwojkowyString, 2)` oraz `Integer.toBinaryString(integerDziesietny)`.

- Napisz funkcję, która w napisie przekazanym jako parametr zamieni wszystkie wystąpienia samogłosek na odpowiadające wielkie litery, a spółgłosek na małe litery. Przyjmijmy, że posługujemy się tylko podstawowym alfabetem łacińskim, bez polskich liter. Ma ona zwracać wynikowy napis.
- Napisz funkcję sprawdzającą, czy przekazane jako parametr słowo składa się tylko z wielkich liter.
- Napisz funkcję **`static int maxIter(String s)`** zwracającą największą z cyfr występujących w liczbie całkowitej `s` obliczoną w sposób iteracyjny.
- Napisz funkcję **`static int maxRek(String s)`** zwracającą największą z cyfr występujących w liczbie całkowitej `s` obliczoną w sposób rekurencyjny.
- Napisz funkcję **`static void zdanie(String nazwaPliku)`** czytającą tekst z podanego pliku i formatującą go tak, aby wielkie litery występowały tylko na początku każdego zdania. Po przetworzeniu zapisany ma zostać nowy plik zawierający w sobie poprawiony tekst. Plik ten ma posiadać taką samą nazwę, jak oryginalny, ale z dopisaną końcówką **`_edited`**. Przykładowo, dla pliku wejściowego „`javaToZycie.txt`”, wyjście powinno nazywać się „`javaToZycie_edited.txt`”.

Wskazówka: możesz wykorzystać klasę `java.io.File`.

3. W metodzie **`main()`** przetestuj obydwie utworzone przez siebie klasy.