

Zadania z programowania w języku Java do wykładu 7 dla II roku Informatyki

Częściowo na podstawie książki: „Java 9. Przewodnik doświadczonego programisty”

Wyrażenia regularne

1. Korzystając z programu `Reth.java` przeanalizuj przykłady ze stron:
<https://docs.oracle.com/javase/tutorial/essential/regex/literals.html>
https://docs.oracle.com/javase/tutorial/essential/regex/char_classes.html
<https://docs.oracle.com/javase/tutorial/essential/regex/quant.html>
2. Przeanalizuj przykłady ze strony
<https://www.w3resource.com/java-exercises/re/index.php>
3. Napisz program, który sprawdza, czy łańcuch zawiera tylko pewien zestaw znaków (w tym przypadku `a-z`, `A-Z` i `0-9`).
4. Napisz program, który dopasowuje łańcuch znaków, zaczynający się od litery `a`, po której występuje zero lub więcej liter `b`.
5. Napisz program, który dopasowuje łańcuch znaków, zaczynający się od litery `a`, po której występuje co najmniej jedna litera `b`.
6. Napisz program, który dopasowuje łańcuch znaków, zaczynający się od litery `a`, po której występują trzy litery `b`.
7. Napisz program, który dopasowuje łańcuch znaków, zaczynający się od litery `a`, po której występuje od dwóch do trzech liter `b`.
8. Napisz program, który znajduje sekwencje małych liter połączone podkreślnikiem.
9. Napisz program, który znajduje łańcuchy zawierające jedną wielką literę, a następnie małe litery.
10. Napisz program, który dopasowuje łańcuch znaków, zaczynający się od litery `a`, po której występuje dowolny ciąg znaków kończący się literą `b`.
11. Napisz program, który dopasowuje podane *słowo* na początku łańcucha. (Przez *słowo* rozumiemy ciąg znaków alfanumerycznych.)
12. Napisz program, który dopasowuje *słowo* na końcu łańcucha, z opcjonalną interpunkcją.
13. Napisz program, który znajduje w łańcuchu *słowa* zawierające literę `Z`.
14. Napisz program, który znajduje w łańcuchu *słowa* zawierające `Z`, ale nie na początku lub końcu *słowa*.
15. Napisz program, który znajduje w łańcuchu podłańcuch zawierający tylko wielkie i małe litery, cyfry oraz znaki podkreślenia.
16. Napisz program, który dopasowuje łańcuch, gdy zaczyna się on od określonej cyfry.
17. Napisz wyrażenie regularne, które sprawdza czy liczba zmiennoprzecinkowa podana przez użytkownika ma poprawny format. Przykładowo, liczby `123`, `456654` czy `-10` są poprawne ale `64-21` czy `\2022`, już nie.

18. Napisz wyrażenie regularne, które sprawdza czy numer domu jest w formacie `numer\numer`. Poprawnym numerem jest `123\2A`, `24B\3` czy `12\5`, ale już numer `abc\cba` nie.
19. Napisz wyrażenie regularne, które sprawdza czy użytkownik wprowadził poprawną nazwę miasta. Przykładowo, `Częstochowa`, `Jelenia Góra` czy `Bielsko-Biała` są poprawnymi nazwami miast, jednak `Myszków123` już nie.
20. Napisz program, który po wczytaniu łańcucha znaków wypisz w kolejnych liniach wszystkie występujące w tym łańcuchu trzyliterowe ciągi znaków zaczynające się od litery `r`. Przykładowo, po wczytaniu łańcucha `I saw a terrible rat running through the forest.`
program powinien wypisać następujące trzyliterowe ciągi: `rri rat run rou res`.
21. Napisz wyrażenie regularne, które sprawdza czy liczba zmiennoprzecinkowa podana przez użytkownika ma poprawny format. Przykładowo, liczby `123`, `456654` czy `-10` są poprawne ale `64-21` czy `\2022`, już nie.
22. Napisz wyrażenie regularne, które sprawdza czy numer domu jest w formacie `numer\numer`. Poprawnym numerem jest `123\2A`, `24B\3` czy `12\5`, ale już numer `abc\cba` nie.
23. Napisz program, który po wczytaniu łańcucha znaków zastąpi w nim wszystkie trzyliterowe słowa zaczynające się od litery `c` na ich odpowiedniki pisane wielkimi literami. Przykładowo, po wczytaniu łańcucha `cat camera can pen cow cab cot`
program powinien wypisać `CAT camera CAN pen COW CAB COT`.
24. Zmodyfikuj program `HrefMatch.java` w taki sposób aby znajdował na podanej jako argument programu stronie internetowej nazwy plików z rozszerzeniem `.jpg`.
25. Korzystając z wyrażenia regularnego, wyodrębnij wszystkie dziesiętne liczby całkowite (łącznie z ujemnymi) z ciągu znaków do zmiennej typu `ArrayList<Integer>`
 - (a) za pomocą metody `find`,
 - (b) za pomocą metody `split`.

Zauważ, że znak `+` lub `-`, po którym nie ma cyfry, oddziela dane.
26. Korzystając z wyrażeń regularnych, wyodrębnij nazwy katalogów (do tablicy ciągów znaków), nazwę pliku i rozszerzenie pliku z bezwzględnej lub względnej ścieżki, takiej jak `/home/student/mójplik.txt`.
27. Ile grup występuje w następującym wyrażeniu regularnym:
`String r = "\\b((\\d3)(\\d3)(\\d4))|((\\d3)(\\d4))\\b";`
28. Podaj rzeczywiste zastosowanie referencji do grup w `Matcher.replaceAll` i zaimplementuj je.