Wyrażenia regularne

Validator

1. Napisz aplikację sprawdzającą czy wprowadzony przez użytkownika numer telefonu jest poprawny. Przygotuj klasę o nazwie `TelephoneValidator` posiadającą metodę

```
public boolean validate(String telephone);
```

   Podany telefon uważa się za poprawny gdy składa się z 9 cyfr.
   Poprawny numer telefonu to: 505879357, 505 879 357, 505-879-357
   Błędny numer telefonu to: 50 58 79 35 7, 5058794, 44505879357

2. Rozbuduj aplikację z poprzedniego zadania tak aby mechanizm sprawdzający weryfikował też numer kierunkowy kraju. Przykładowo numer telefonu +48505888159 jak i 505888159 jest poprawny.

3. Przygotuj aplikację sprawdzającą czy wprowadzone imię jest poprawne. Użytkownik wprowadza tekst w postaci “Jan”, a następnie program sprawdza jego poprawność. Przygotuj klasę o nazwie NameValidator posiadającą metodę
   `public boolean validate(String name);`
   Poprawne imię to: Jan, Monika, Łukasz, Krzysztof
   Błędne imię to: jan, monika, jan87, Jan87
   Imię uważa się za poprawne, gdy zawiera tylko litery i rozpoczyna się wielką literą.

4. Przygotuj aplikację sprawdzającą czy podany numer tablicy rejestracyjnej jest poprawny. Przygotuj klasę o nazwie PlateValidator posiadającą metodę
   `public boolean validate(String plate);`
   Na potrzeby zadania przyjmij, że tablica rejestracyjną jest poprawna gdy zawiera 2 litery, a następnie 5 cyfr lub 4 cyfry i jedna litery lub 3 cyfry i 2 litery.
   Poprawny numer tablicy rejestracyjnej to: CB3456J, CB34212, WY640WI
   Błędny numer tablicy rejestracyjnej to: CBS3456, W1234YU, CC14WYG

5. * Przygotuj aplikację sprawdzającą czy wprowadzony przez użytkownika adres www jest poprawny
   `public boolean validate(String www);`
   Poprawny adres www to: www.wp.pl
   Błędny adres www to: http://, http://-error-.invalid/, http://foo.bar/foo(bar)baz quux, http:///www.wp.pl

6. * Przygotuj aplikację sprawdzającą czy wprowadzony przez użytkownika adres e-mail jest poprawny. Przygotuj klasę o nazwie EmailValidator posiadającą metodę
   `public boolean validate(String email);`
   Przygotuj test jednostkowy sprawdzający poprawność działania.

Poprawny e-mail	Błędny e-mail
prettyandsimple@example.com very.common@example.com disposable.style.email.with+symbol@ex	john.doe@example..com Abc.example.com just"not"right@example.com

ample.com other.email-with-dash@example.com	john..doe@example.com

7. * Przygotuj aplikację sprawdzającą czy wprowadzony przez użytkownika adres IP jest poprawny.
   `public boolean validate(String address);`
   Poprawny adres IP: 192.168.1.10, 10.10.48.1
   Błędny adres IP: 192.168.256.20, 192.168.2,

8. * Przygotuj aplikację weryfikującą czy wprowadzone przez użytkownika hasło jest poprawne.
   Założenia dotyczące hasła:
   - Hasło składa składa się co najmniej 8 i nie więcej niż 16 znaków
   - Hasło zawiera co najmniej 2 małe literę.
   - Hasło zawiera co najmniej 2 duże literę.
   - Hasło zawiera co najmniej jedną cyfrę
   - Hasło zawiera co najmniej jeden znak specjalny

Parser

9. * Przygotuj aplikację, która pobiera od użytkownika ścieżkę do pliku HTML, otwiera go i pobiera wszystkie obrazy w nim zawarte.

10. * Przygotuj aplikację, która pobiera od użytkownika ścieżkę do pliku HTML, otwiera go i pobiera wszystkie adresy www w nim zawarte.

11. * Napisz aplikację, która wczytuje plik CSV, a następnie otwiera go i pobiera z niego imię, nazwisko, adres e-mail oraz wiek. Utwórz klasę Person przechowującą wszystkie pobrane parametry i umieść wszystkie osoby na liście.