



Universidade Federal da Paraíba - UFPB

Rennan Wesley da Silva Costa

Teoria da Computação – Projeto 1

João Pessoa – PB

Abril 2018

O projeto foi feito utilizando Expressões Regulares através de uma linguagem de programação (**Java**).

Uma expressão regular em Java é uma **String** formatada que determina um padrão que deve ser seguido, esta formatação ocorre através dos **Metacaracteres**, **Modificadores**, **Quantificadores** e **Agrupadores**.

Das ferramentas de formatação citadas acima, apenas os Modificadores que não foram utilizados.

Metacaracteres (inclusive os de Fronteira):

- \d: Busca por qualquer dígito de 0 até 9;
- "|": É o operador de união (**ou**).

Quantificadores:

- A{n}: Busca n vezes a ocorrência do caractere A.

Agrupadores:

- [A-Z]: Agrupador de alcance, procura por qualquer letra (maiúscula) entre A e Z. Diferencia letras maiúsculas das minúsculas (para letras minúsculas seria [a-z]).

Até então sabe-se como determinar a expressão regular, mas a leitura da cadeia em si é feita através do método, da classe String, **matches**. O método tem como função decidir se a cadeia vai ou não ser aceita pela expressão, para isso, é retornado **true** caso a cadeia seja aceita e **false**, caso contrário.

Com isso, já se é possível juntar as peças e fazer o projeto em si.

Já que se tratam de três possíveis casos, pode-se criar uma expressão regular para cada um e no final uni-las, através do operador de união “|” (ou), em uma só expressão. Assim feito, a cadeia será aceita se pelo menos uma expressão regular dos três casos a aceitarem.

Expressões:

- CPF: $([0 - 9]^3 \cdot [0 - 9]^3 \cdot [0 - 9]^3 - [0 - 9]^2)$;
- Telefone: $(([0 - 9]^4 \text{ ou } [0 - 9]^5) - [0 - 9]^4)$;
- Placa de Carro: $([A - Z]^3 - [0 - 9]^4)$;
- Final: (CPF | Telefone | Placa de Carro).

Transferindo as expressões dos três casos para Java:

```
String cpf = "\\d{3}\\.\\d{3}\\.\\d{3}-\\d{2}",  
num = "\\d{4}-\\d{4}|\\d{5}-\\d{4}",  
pla = "[A-Z]{3}-\\d{4}";
```

O programa testa as cadeias em um laço **for** através do array de strings **s**. O método `matches` é chamado dentro do laço em um operador de decisão **if**, onde é comparado o seu valor de retorno. Se `matches` retornar `true` o programa exibe que a cadeia foi aceita e exibe que a cadeia foi rejeitada, caso `false`.

```
for(String o : s)  
    if(o.matches(cpf + "|" + num + "|" + pla))  
        System.out.println(o + ": Accepted.\n");  
    else  
        System.out.println(o + ": Rejected.\n");
```

Código completo: [www.github.com/RennanWeslley/Teoria](https://github.com/RennanWeslley/Teoria)

Testes

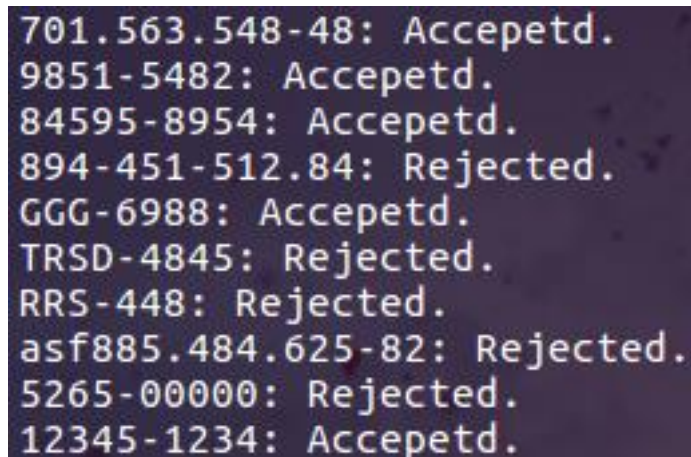
Como dito antes, os testes são feitos através de um array de strings s. Foram testadas 10 cadeias, segue um trecho do código e a sua execução:

Casos:

```
String[] s = new String[10];

s[0] = "701.563.548-48";
s[1] = "9851-5482";
s[2] = "84595-8954";
s[3] = "894-451-512.84";
s[4] = "GGG-6988";
s[5] = "TRSD-4845";
s[6] = "RRS-448";
s[7] = "asf885.484.625-82";
s[8] = "5265-00000";
s[9] = "12345-1234";
```

Execução:



```
701.563.548-48: Accepethd.
9851-5482: Accepethd.
84595-8954: Accepethd.
894-451-512.84: Rejected.
GGG-6988: Accepethd.
TRSD-4845: Rejected.
RRS-448: Rejected.
asf885.484.625-82: Rejected.
5265-00000: Rejected.
12345-1234: Accepethd.
```