Review - matches

\\ backslash character

To use a predefined character, it must be preceded by a \ as follows: "abc".matches("\\w+");

**Predefined character classes**

. Any character (may or may not match a line terminator)

\d A digit: [0-9]

\D A non-digit: [^0-9]

\s A whitespace character: [ \t\n\x0B\f\r]

\S A non-whitespace character: [^\s]

\w A word character: [a-zA-Z\_0-9]

\W A non-word character: [^\w]

**Character Classes**

[abc] a, b, or c (simple class)

[^abc] Any character except a, b, or c (negation)

[a-zA-Z] a through z or A through Z, inclusive (range)

[a-d[m-p]] a through d, or m through p: [a-dm-p] (union)

[a-z&&[def]] d, e, or f (intersection)

[a-z&&[^bc]] a through z, except for b and c: [ad-z] (subtraction)

[a-z&&[^m-p]] a through z, and not m through p: [a-lq-z](subtraction)

Greedy quantifiers

X? *X*, once or not at all

X\* *X*, zero or more times

X+ *X*, one or more times

X{n} *X*, exactly *n* times

X{n,} *X*, at least *n* times

X{n,m} *X*, at least *n* but not more than *m* times

Review #18 - matches

String s = "UIL";

1. out.println(s.matches(".+U.\*"));

2. out.println(s.matches(".\*U.+"));

3. out.println(s.matches(".\*U.\*"));

4. out.println(s.matches(".+U.+"));

5. out.println(s.matches(".+[UL].+"));

String t = "COMPUTER";

6. out.println(t.matches(".+U.\*"));

7. out.println(t.matches(".\*U.+"));

8. out.println(t.matches(".\*U.\*"));

9. out.println(t.matches(".+U.+"));

10. out.println(t.matches(".+[UL].+"));

String z = "SCIENCE 101";

11. out.println(z.matches("..+I.\*"));

12. out.println(z.matches("..\*IE.+"));

13. out.println(z.matches(".^I.\*"));

14. out.println(z.matches(".[^IE].\*"));

15. out.println(z.matches(".[^IEC].\*"));

String h = "first\_last@my-address3.com";

16. out.println(h.matches("\\w+\_\\w+@\\w\*-\\.\\w\*"));

17. out.println(h.matches("\\w+\_\\w+@\\w\*-\\w+\\.\\w\*"));

18. out.println(h.matches(".+@.\*\\d\\.com"));

19. out.println(h.matches(".\\D+.\*@\\.com"));

20. out.println(h.matches("\\w\*@\\D\*\\d\\.\\w+"));

String j = "-1,300,000.02", k = "7777.142857";

21. out.println(j.matches("-?\\d{1,3}(,\\d{3})\*(\\.\\d+)?"));

22. out.println(j.matches("-\\d{1,3}(,\\d{3})\*(\\.\\d+)"));

23. out.println(j.matches("-\\d{1},\\d{3}+\\.\\d+"));

24. out.println(k.matches("\\d{1,4}(\\.\\d+)"));

25. out.println(k.matches("\\d{4}(\\.\\d?)"));

String p = "AA11BB22CC33", q = "A-1-BB-22-CCC-333";;

26. out.println(p.matches("\\w\*"));

27. out.println(p.matches("(\\w{2}\\d{2})\*"));

28. out.println(p.matches("\\w?"));

29. out.println(q.matches("(\\w{1,3}-\\d{1,3}-?)+"));

30. out.println(q.matches("(\\w{1,3}-\\d{1,3}-+)+"));1. \_\_\_\_\_\_\_\_\_

2. \_\_\_\_\_\_\_\_\_

3. \_\_\_\_\_\_\_\_\_

4. \_\_\_\_\_\_\_\_\_

5. \_\_\_\_\_\_\_\_\_

6. \_\_\_\_\_\_\_\_\_

7. \_\_\_\_\_\_\_\_\_

8. \_\_\_\_\_\_\_\_\_

9. \_\_\_\_\_\_\_\_\_

10. \_\_\_\_\_\_\_\_\_

11. \_\_\_\_\_\_\_\_\_

12. \_\_\_\_\_\_\_\_\_

13. \_\_\_\_\_\_\_\_\_

14. \_\_\_\_\_\_\_\_\_

15. \_\_\_\_\_\_\_\_\_

16. \_\_\_\_\_\_\_\_\_

17. \_\_\_\_\_\_\_\_\_

18. \_\_\_\_\_\_\_\_\_

19. \_\_\_\_\_\_\_\_\_

20. \_\_\_\_\_\_\_\_\_

21. \_\_\_\_\_\_\_\_\_

22. \_\_\_\_\_\_\_\_\_

23. \_\_\_\_\_\_\_\_\_

24. \_\_\_\_\_\_\_\_\_

25. \_\_\_\_\_\_\_\_\_

26. \_\_\_\_\_\_\_\_\_

27. \_\_\_\_\_\_\_\_\_

28. \_\_\_\_\_\_\_\_\_

29. \_\_\_\_\_\_\_\_\_

30. \_\_\_\_\_\_\_\_\_