

Metody na Stringach:

1. **concat()**: Metoda `concat()` służy do łączenia dwóch stringów, tworząc nowy string zawierający połączone wartości.
2. **length()**: Ta metoda zwraca długość (liczbę znaków) stringa.
3. **charAt()**: Pozwala uzyskać znak na określonej pozycji w stringu. Możesz wywołać `charAt(index)` i przekazać numer pozycji znaku, który chcesz uzyskać.
4. **substring()**: Metoda `substring(beginIndex, endIndex)` zwraca podciąg stringa od `beginIndex` do `endIndex - 1`.
5. **toLowerCase()** i **toUpperCase()**: Pozwalają na zmianę wielkości liter w stringu. `toLowerCase()` zamienia wszystkie litery na małe, a `toUpperCase()` na duże.
6. **trim()**: Usuwa białe znaki (spacje) z początku i końca stringa, pomagając oczyścić dane.
7. **split()**: Pozwala podzielić string na tablicę stringów na podstawie określonego separatora.
8. **equals()** i **equalsIgnoreCase()**: `equals()` porównuje dwa stringi pod kątem dokładnego dopasowania, a `equalsIgnoreCase()` robi to samo, ignorując wielkość liter.
9. **indexOf()** i **lastIndexOf()**: Pozwalają znaleźć pozycję pierwszego lub ostatniego wystąpienia danego znaku lub ciągu znaków w stringu.
10. **replace()**: Służy do zamiany wszystkich wystąpień określonego ciągu znaków na inny ciąg w stringu.
11. **startsWith()** i **endsWith()**: Sprawdzają, czy string zaczyna się lub kończy określonym ciągiem znaków.
12. **contains()**: Sprawdza, czy dany ciąg znaków występuje w stringu.
13. **format()**: Metoda `String.format()` pozwala formatować stringi zgodnie z podanymi specyfikacjami i zastępowaniem wartości.
14. **StringBuilder** i **StringBuffer**: Klasy te pozwalają na efektywne tworzenie i modyfikowanie stringów, szczególnie w przypadku operacji, które generują wiele pośrednich ciągów znaków.
15. **isEmpty()**: Metoda ta sprawdza, czy string jest pusty (nie zawiera znaków).

Metody z klasy Math:

1. `abs(double a)`: Metoda ta zwraca wartość bezwzględną podanej liczby `a`. To oznacza, że zawsze zwraca wartość nieujemną, niezależnie od znaku liczby.
2. `sqrt(double a)`: Ta metoda zwraca pierwiastek kwadratowy liczby `a`. Wynik jest liczbą zmiennoprzecinkową.
3. `pow(double a, double b)`: Służy do podnoszenia liczby `a` do potęgi `b`. Metoda zwraca `a` podniesione do potęgi `b` jako wartość zmiennoprzecinkową.
4. `exp(double a)`: Zwraca wartość eksponenty, tj. e^a , gdzie e to liczba Eulera, a `a` to argument.
5. `log(double a)`: Oblicza naturalny logarytm z liczby `a`, czyli logarytm o podstawie e .
6. `max(double a, double b)` i `min(double a, double b)`: Te dwie metody pozwalają znaleźć maksimum i minimum spośród dwóch liczb `a` i `b`.
7. `ceil(double a)` i `floor(double a)`: `ceil` zaokrągla liczbę `a` w górę do najbliższej liczby całkowitej, a `floor` zaokrągla w dół.
8. `round(double a)`: Zaokrągla liczbę `a` do najbliższej liczby całkowitej.
9. `random()`: Generuje losową liczbę z zakresu $[0, 1)$. Możesz mnożyć tę wartość przez dowolny inny zakres, aby uzyskać losową liczbę w danym przedziale.
10. `sin(double a)`, `cos(double a)`, `tan(double a)`: Te metody obliczają odpowiednio funkcje sinus, cosinus i tangens kąta `a`, gdzie `a` jest podane w radianach.
11. `toDegrees(double radians)` i `toRadians(double degrees)`: Te metody konwertują między radianami a stopniami.