

Лабораторна робота № 8

РОЗРОБКА ЗАСТОСУНКІВ МОВОЮ ПРОГРАМУВАННЯ JAVA. РОБОТА ІЗ РЯДКАМИ СИМВОЛІВ

Мета роботи: отримання практичних навичок роботи з рядками символів в мові програмування Java: String.

Досліджуваний матеріал: Рядок символів - це впорядкована послідовність символів. Рядок є основним носієм інформації. Для роботи зі рядком символів використовуються класи: String, StringBuilder, StringBuffer. На цій лабораторній роботі вивчимо клас String.

1. Постановка задачі

Навчитися працювати з класом String.

2. Порядок виконання роботи

1. Прочитати теоретичний матеріал
2. Вивчити роботу оператора if-else
3. Вивчити роботу оператора switch-case
4. Вирішити завдання з використанням вивченого матеріалу
5. Зробити висновки по роботі.

3. Зміст звіту

1. Постановка завдання.
2. Встановити завдання, які вирішуються в лабораторній роботі
3. Коротко описати роботу класу String.
4. Записати код вирішення завдань.
5. Висновки по роботі.

4. Теоретичні відомості

4.1. Клас String

Тип String не є примітивним типом даних, однак це один з найбільш використовуваних типів в Java. String призначений для зберігання рядків тексту.

```
String a1;
```

```
String a1= new String ();
```

```
String a2="";
```

```
String a3="об'єкт тип String");
```

```
String a4= new String ("");
```

```
String a5= new String (" об'єкт тип String");
```

```
String name1 = "Vasya";  
String name2 = "Vasya";  
String name3 = new String("Vasya");  
System.out.println(name1 == name2);      true  
System.out.println(name1 == name3);      false  
System.out.println(name1.equals(name3)); true
```

```
String a1= "Один";
```

```
String a2= "Два";
```

```
String a3= a1 + " і " + a2
```

```
System.out.println(a1);
```

```
System.out.println(a2);
```

```
System.out.println(a3);
```

```
char b1[ ]=new char [a1.length()];
```

```
b1[1]=a1.charAt(1);
```

```
String str[] = { "один", "два", "три" };  
for(int i=0; i<str.length;i++)  
System.out.println("str[" + i + "]: " + str[ i]);
```

```
char c1[] = { 'a', 'b', 'c' };  
String s1 = new String(c1);  
System.out.println(s1);
```

Масив типу String

```
String[] we = {"Я", "Ты", "Он", "Она"};
```

Слово важке, але потрібно його запам'ятати (Конкатенація). Для того, щоб об'єднати кілька різних рядків в одну, в Java можна використовувати перевантажені (спеціально для об'єктів String) оператори «+» і «= +».

Ми це робили багато разів в процесі виведення даних за допомогою

```
String[] we = {"Я", "Ты", "Он", "Она"};  
String then = we[1] + ", " + we[2] + ", " + we[3];  
System.out.println(we[0] + ", " + we[1]);  
System.out.println(then);
```

4.2. Найбільш використовувані методи класу String

Визначення кількості символів в рядку

Для того щоб визначити кількість символів в рядку, використовується метод length.

int length() — повертає довжину рядка. Довжина дорівнює кількості символів Unicode в рядку (рис.1).

```
1 package as;
2
3 public class As {
4
5     public static void main(String[] args) {
6
7         String[] we = {"Я", "Ты", "Он", "Она"};
8         String then = we[1] + ", " + we[2] + ", " + we[3];
9         System.out.println(we[0] + ", " + we[1]);
10        System.out.println(then);
11        System.out.println(then.length());
12    }
13 }
```

Problems @ Javadoc Declaration Console

<terminated> As [Java Application] C:\Program Files (x86)\Java\jre1.8.0_181\bin\javaw.exe (1 вер. 2018 р., 19:12:)

Я, Ты
Ты, Он, Она
11

Рисунок 1 – Використання методу length()

Отримання символу з рядка за номером (індексом)

Якщо нам необхідний символ з рядка з конкретної позиції, можна використовувати метод charAt.

char charAt(int index) — повертає символ, що знаходиться за вказаною індексу в рядку. Результатом роботи методу буде символ типу char. Параметр, який передається в метод - ціле число.

```
7     String[] we = {"Я", "Ты", "Он", "Она"};
8     String then = we[1] + ", " + we[2] + ", " + we[3];
9     System.out.println(we[0] + ", " + we[1]);
10    System.out.println(then);
11    System.out.println(then.charAt(1));
12 }
```

Problems @ Javadoc Declaration Console

<terminated> As [Java Application] C:\Program Files (x86)\Java\jre1.8.0_181\bin\javaw.exe (1 вер. 2018 р., 19:20:)

Я, Ты
Ты, Он, Она
Ы

Рисунок 2 – Використання методу charAt()

Витяг підрядка з рядка

`String substring(int beginIndex, int endIndex)` або `substring(int beginIndex)` — повертає новий рядок, який є підрядком використаного рядку. У параметрах методу потрібно вказати індекс рядка, з якого починається підрядок символів та індекс, яким закінчується. Також можливо вказувати тільки початковий індекс. У цьому випадку буде повернуто підстрока символів від початкового індексу і до кінця рядка.

🌟 `substring(int beginIndex) : String` - String - 0,05%
🌟 `substring(int beginIndex, int endIndex) : String` - String - 0,05%
🟢 `subSequence(int beginIndex, int endIndex) : CharSequence` - String
substring to Code Reco
`public String substring(int beginIndex)`
"unhappy".substring(2) returns "happy"
"Harbison".substring(3) returns "bison"
"emptiness".substring(9) returns "" (an empty string)

```
7      String[] we = {"0", "12", "345", "5678"};  
8      String then = we[0] + we[1] + we[2] + we[3];  
9      System.out.println(then);  
10     System.out.println(then.substring(2,4));  
11     System.out.println(then.substring(2));
```

Problems @ Javadoc Declaration Console

<terminated> As [Java Application] C:\Program Files (x86)\Java\jre1.8.0_181\bin\javaw.exe (1 вер. 2018 р., 19:35:28)

0123455678
23
23455678

Рисунок 3 – Використання методу `substring`

Розбиття рядків

Для розбиття рядків на частини використовується метод `String [] split (String regex)`. Рядок символів видається методом у вигляді масиву рядків. А масив формується на основі ознаки всередині дужок методу.

```
5 public static void main(String[] args) {
6
7     String[] we = {"0", "12", "345", "5678"};
8     String then = we[0] + we[1] + we[2] + we[3];
9     System.out.println(then);
10    String []splitMethod=then.split("34");
11    System.out.println(splitMethod[0]);
12    System.out.println(splitMethod[1]);
13
14
```

Problems @ Javadoc Declaration Console

<terminated> As [Java Application] C:\Program Files (x86)\Java\jre1.8.0_181\bin\javaw.exe (1 вер. 2018 р., 1
0123455678
012
55678

Рисунок 4 – Використання методу split

Порівняння рядків

boolean equals(Object anObject) — перевіряє ідентичність рядків. Повертає true тільки в тому випадку, якщо в рядку символів представлена однакова послідовність символів однієї величини.

```
7     String[] we = {"0", "12", "345", "5678"};
8     String then = we[0] + we[1] + we[2] + we[3];
9
10    System.out.println(then.equals("0123455678"));
--
```

Problems @ Javadoc Declaration Console

<terminated> As [Java Application] C:\Program Files (x86)\Java\jre1.8.0_181\bin\javaw.exe (1 вер. 2018 р., 20:06
true

Рисунок 4 – Використання методу equals

4.3. Інші методи класу String

`boolean contains(CharSequence s)` — перевіряє, чи містить рядок символів задану послідовність символів і повертає `true` або `false`.

`boolean endsWith(String suffix)` — перевіряє завершується чи рядок символів певними символами і повертає `true` або `false`.

`boolean startsWith(String prefix)` або `startsWith(String prefix, int toffset)` — перевіряє, чи починається рядок символів з певних символів. У другому випадку можна вказати позицію з якої необхідно почати пошук префікса.

`int indexOf(int ch)`, `indexOf(int ch, int fromIndex)`, `indexOf(String str)`, `indexOf(String str, int fromIndex)` — метод `indexOf` застосовується для пошуку першого входження зазначеного символу в рядку або першого входження зазначеної підрядка. Пошук також можна зробити із зазначенням позиції в рядку від якого потрібно починати шукати. Для пошуку потрібно вказати відповідні параметри. Метод повертає число відповідне індексу першого входження символу або підрядка. У разі відсутності зазначеного символу або підрядка в рядку, буде повернута `-1`.

Модифікація рядків не є модифікацією як такої. Справа в тому, що об'єкти класу `String` після створення вже не можна змінювати. Але можна створити копію рядка зі змінами. Саме це і роблять такі методи.

`toLowerCase()` — перетворити рядок в нижній регістр;

`toUpperCase()` — перетворити рядок у верхній регістр;

`trim()` — відсікти на кінцях рядки порожні символи;

`String replace(char oldChar, char newChar)`, `replace(CharSequence target, CharSequence replacement)` — заміна в рядку одного символу або підрядка на інший символ або підрядок.

`int compareTo(String anotherString)` — так само перевіряє ідентичність рядків, однак, на відміну від методу `equals` повертає: нульове значення, якщо рядки рівні, ціле негативне число, якщо перший рядок символів передре другий, ціле позитивне число, якщо перший рядок символів слід за другим

Клас String має багато методів (десь 40). Інші методи можна (а якщо ви збираєтеся стати програмістами, то і ПОТРІБНО) подивитися в офіційній документації <https://docs.oracle.com/javase/8/docs/api/java/lang/String.html#methodsummary>

Нижче наведені завдання для самостійної роботи. Я впевнений, що той хто зможе вирішити ці завдання стає програмістом.

Висновок

Отримано практичні навички роботи з операторами з класу String мови програмування Java.

5. Контрольні питання

1. Особливості роботи з класом String?
2. Розповісти про роботу з обраними методами класу String?

Завдання

1. Дано рядок символів. Вивести його три рази через кому і показати кількість символів в ньому.
2. Дано рядок символів. Вивести перший, останній і середній (якщо він є) символи.
3. Дано рядок символів. Вивести перші три символи і останні три символи, якщо довжина рядка більше 5. Інакше вивести перший символ стільки разів, яка довжина рядка.
4. Сформувати рядок з 10 символів. На парних позиціях повинні знаходитися парні цифри, на непарних позиціях - літери.
5. Дано рядок символів. Показати номери символів, які збігаються з останнім символом рядка.
6. Дано рядок символів. Показати третій, шостий, дев'ятий і так далі символи.
7. Дано рядок символів. Визначте загальну кількість символів '+' і '-' в ньому. А також скільки таких символів, після яких слід цифра нуль.
8. Дано рядок символів. Визначте, який символ в ній зустрічається раніше: 'x' або 'w'. Якщо якогось із символів немає, вивести повідомлення про це.

9. Дано два рядки. Вивести більший по довжині рядок стільки разів, на скільки символів відрізняються рядки.

10. Дано рядок символів. Якщо він починається на 'abc', то замінити їх на 'www', інакше додати в кінець рядка 'zzz'.

11. Дано рядок символів. Якщо його довжина більше 10, то залишити в рядку тільки перші 6 символів, інакше доповнити рядок символами 'o' до довжини 12.

12. Дано рядок символів. Розділити рядок на фрагменти по три поспіль символу. У кожному фрагменті середній символ замінити на випадковий символ, який не збігається ні з одним з символів цього фрагмента. Показати фрагменти, впорядковані за алфавітом.

13. Дано рядок символів. Замінити кожен парний символ або на 'a', якщо символ НЕ дорівнює 'a' або 'b', або на 'c' в іншому випадку.

14. Дано рядок символів. У рядку знайти кількість цифр.

15. Дано рядок символів. Визначити, чи містить рядок символів тільки символи 'a', 'b', 'c' чи ні.

16. Дано рядок символів. Замініть в рядку всі входження 'word' на 'letter'.

17. Дано рядок символів. Видаліть в рядку всі літери 'x'. за якими слід 'abc'.

18. Дано рядок символів. Видаліть в рядку всі 'abc', за якими слід цифра.

19. Дано рядок символів. Знайдіть кількість входження 'aba' в рядок.

20. Дано рядок символів. Видалити в рядку всі зайві прогалини, тобто серії поспіль прогалин замінити на поодинокі прогалини. Крайні прогалини в рядку видалити.

21. Дано рядок символів, що складається з слів, розділених символами, які перераховані у другому рядку. Показати всі слова.

22. Дано рядок символів. Знайдіть найбільшу кількість підряд цифр.

23. Дано рядок символів. Знайти суму наявних в ньому цифр.

24. Дано рядок символів. Знайти слова, що складаються з цифр, і суму чисел, які утворюють ці слова.

25. Дано рядок символів. Знайдіть найбільшу кількість поспіль прогалин в ньому.

26. Дано два слова. Знайдіть тільки ті символи слів, які зустрічаються в обох словах лише один раз.

27. Дано масив рядків. Впорядкувати масив по довжині рядків.

28. Дано масив рядків. Переставити рядки в залежності від кількості цифр в рядку.

29. Дано рядок символів. Замінити всі символи 'a' і 'b' на 'A' і 'B' відповідно.

30. Дано рядок символів. Сформувати рядок із символів, розташованих між першим і другим запитом даного тексту.