РЯДКИ

Рядки

Рядок – послідовність символів Юнікод

На примітивному рівні рядки – масив типу char

В Java рядки описуються як об’єкти класу String

Спецсимволи

Керуюча конструкція	Описання
\t	Символ табуляції.
\b	Символ повернення в тексті на один крок назад або видалення одного символа в стрічці (backspace).
\n	Символ перехода на нову стрічку.
\r	Символ повернення каретки.
\f	Прогон сторінки.
\'	Символ одинарної лапки.
\"	Символ подвійної лапки.
\\	Символ зворотньої косої риски(\).

Конкантенація

```
String str = "Це рядок";
String str = "Це"+" рядок";
```

- Конкантенація з іншими типами

```
String str3 = "цифра " + 5;
//String + int дає String "цифра 5«
```

- Метод concat():

```
String strEnd = "рулить";
String str = "Java ".concat(strEnd);
//в результаті str="Java рулить"
```

Робота з рядками

```
String str = "Це";
String str2 = "рядок";
String str3 = "555";
str = str3; //так можна
str = str + " " + str2;   //і так можна
```

- Підрядки

substring(pos1, pos2)

```
String greeting = "Hello";
String s = greeting.substring(0,3); //скопіювати з greeting символи з 0 до 3, тобто 0, 1 та 2 – “Hel”
```

Порівняння рядків

- Порівняння двох строкових змінних

```
str1.equals(str2);
```

- Порівняння строкової змінної і константи

```
str1.equals(«Hello»);
```

- Порівняння без урахування регістру

```
"Hello".equalsIgnoreCase("hello");
```

Приведення до типу

```
import java.util.Scanner;
 public class TestValueOf {
    public static void main(String[] args) {
      int intNumber = 36;

      System.out.print("Введіть число: ");
      //зчитуємо число з клавіатури
      Scanner in = new Scanner(System.in);
      String doubleStr = in.next();
      System.out.println("Ви ввели: " + doubleStr);
      //Ціла і дробова частина повинна бути через крапку. Шукаємо чи не ввели через кому
      int index = doubleStr.indexOf(",");

      if (index >= 0) {
         System.out.println("Кома у позиції: " + index);
         doubleStr = doubleStr.replace(',', '.'); //замінити кому на крапку
      }
      String strNumber = String.valueOf(intNumber); //Перетворюємо int число у рядок тексту
      String strOut = "*".concat(strNumber) + "="; //Приєднуємо число до рядка через метод concat
      double number = Double.valueOf(doubleStr); //Перетворюємо введений рядок тексту у число
      number = number * intNumber; //множимо введене число на 36
      System.out.println(doubleStr + strOut + number);
```

Результат виконання:

```
Введіть число: 555,5
Ви ввели: 555,5
Кома у позиції: 3
555.5*36=19998.0
```

String API

Тип повернення	Назва методу та його аргументи	Опис
char	charAt(int index)	Повертає char значення за вказаним індексом
int	codePointAt(int index)	Повертає символ (Unicode code point) за вказаним індексом
int	codePointBefore(int index)	Повертає символ (Unicode code point) перед вказаним індексом
int	codePointCount(int beginIndex, int endIndex)	Повертає кількість кодових точок Unicode у зазначеному інтервалі в рядку.
int	compareTo(String anotherString)	Порівнює два рядки лексикографічно
int	compareToIgnoreCase(String str)	Порівнює два рядки лексикографічно, ігноруючи різницю в регістрах літер
String	concat(String str)	приєднує зазначений рядок str в кінець рядка
boolean	contains(CharSequence s)	Повертає true, тільки якщо рядок містить зазначену послідовність значень char
boolean	contentEquals(CharSequence cs)	Порівнює рядок із зазначеною послідовністю символів(CharSequence)
boolean	endsWith(String suffix)	Перевіряє чи рядок закінчується зазначеним суфіксом
boolean	equals(Object anObject)	Порівнює рядок із зазначеним об'єктом
boolean	equalsIgnoreCase(String anotherString)	Порівнює рядок з іншим рядком, ігноруючи регістр
byte[]	getBytes()	Кодує рядок у послідовність байт використовуючи символьний набір (charset) по замовчуванню, результат зберігається у новому байтовому масиві
byte[]	getBytes(Charset charset)	Кодує рядок у послідовність байт використовуючи наданий символьний набір(charset), результат зберігається у новий байтовий масив
byte[]	getBytes(String charsetName)	кодує рядок у послідовність байт використовуючи названий символьний набір, результат зберігається у новому байтовому масиві
void	getChars(int srcBegin, int srcEnd, char[] dst, int dstBegin)	Копіює символи з рядка у символьний масив
int	hashCode()	Повертає ХЕШ-код для рядки
int	indexOf(int ch)	Повертає індекс першого входження зазначеного символу в рядок
int	indexOf(int ch, int fromIndex)	Повертає індекс символу у рядку починаючи пошук із зазначеного індексу
int	indexOf(String str)	Повертає індекс першого знаходження підрядка у рядку

boolean	isEmpty()	Повертає true, тільки тоді, якщо довжина(length()) становить 0.
int	lastIndexOf(int ch)	Повертає індекс останнього входження зазначеного символу в рядку
int	lastIndexOf(int ch, int fromIndex)	Повертає індекс останнього входження зазначеного символу, шукаючи його із зазначеної позиції в рядку
int	lastIndexOf(String str)	Повертає індекс останнього входження зазначеного підрядка
int	lastIndexOf(String str, int fromIndex)	Повертає індекс останнього входження зазначеного підрядка, шукаючи його із зазначеного індексу у рядку
int	length()	Повертає довжину даного рядка
boolean	matches(String regex)	Говорить чи відповідає даний рядок заданому регулярному виразу
String	replace(char oldChar, char newChar)	Повертає новий рядок заміняючи усі входження символу(oldChar) в рядку на новий символ (newChar)
String	replace(CharSequence target, CharSequence replacement)	Заміняє в рядку підрядок target новою послідовністю replacement
String	replaceAll(String regex, String replacement)	Заміняє кожен підрядок в рядку, що співпадає з регулярним виразом(regex) новим підрядком(replacement)
String	replaceFirst(String regex, String replacement)	Заміняє перший підрядок, що відповідає заданому регулярному виразу на підрядок для заміни
String[]	split(String regex)	Розбиває рядок за певним правилом поданим у регулярному виразі
String[]	split(String regex, int limit)	Розбиває рядок за певним правилом поданим у регулярному виразі
boolean	startsWith(String prefix)	Перевіряє чи поточний рядок починається з заданого префікса
String	substring(int beginIndex)	Повертає підрядок з поточного рядка
String	substring(int beginIndex, int endIndex)	Повертає підрядок з поточного рядка
char[]	toCharArray()	Перетворює рядок у новий символьний масив
String	toLowerCase()	Перетворює усі символи рядка у нижній регістер використовуючи locale правило по замовчуванню
String	toLowerCase(Locale locale)	Перетворює усі символи рядка у нижній регістр використовуючи правило Locale.
String	toUpperCase()	Конвертує всі символи рядка у верхній регістер використовуючи locale правило по замовчуванню
String	toUpperCase(Locale locale)	Перетворює усі символи рядка у верхній регістер використовуюче правило подане у Locale.
String	trim()	Повертає копію рядка усуваючи пробіли спереду і ззаду рядка
static String	valueOf(boolean b)	Повертає рядкове представлення аргументу boolean типу
static String	valueOf(char c)	Повертає рядкове представлення char аргументу
static String	valueOf(char[] data)	Повертає рядкове представлення масиву типу char
static String	valueOf(double d)	Повертає рядкове представлення double аргументу
static String	valueOf(float f)	Повертає рядкове представлення float аргументу
static String	valueOf(int i)	Повертає рядкове представлення int аргументу.
static String	valueOf(long l)	Повертає рядкове представлення аргументу типу long
static String	valueOf(Object obj)	Повертає представлення об'єкту у вигляді рядка

StringBuilder

Конструктори StringBuilder	
Конструктор	Опис
StringBuilder()	Створює пустий об'єкт StringBuilder з ємністю 16 (16 порожніх елементів).
StringBuilder(CharSequence cs)	Створює об'єкт StringBuilder, що містить ту ж кількість символів як CharSequence, плюс 16 порожніх елементів вкінці.
StringBuilder(int initCapacity)	Створює порожній об'єкт StringBuilder із заданою ємністю.
StringBuilder(String s)	Створює об'єкт StringBuilder, ініціалізований заданим рядком String, плюс 16 додаткових порожніх елементів вкінці

Методи довжини і ємності	
Метод	Опис
void setLength (int newLength)	Встановити довжину послідовності симовлів. Якщо нова довжина (newLength) менше ніж length(), символи, які виходять за межі даної довжити будуть відкинуті. Якщо newLength більша ніж length(), то вкінець послідовності додаються порожні (null) символи.
void ensureCapacity (int minCapacity)	Гарантує, що ємність щонайменше рівна зазначеному мінімуму

Різноманітні методи StringBuilder

Метод	Опис
StringBuilder append(boolean b) StringBuilder append(char c) StringBuilder append(char[] str) StringBuilder append(char[] str, int offset, int len) StringBuilder append(double d) StringBuilder append(float f) StringBuilder append(int i) StringBuilder append(long lng) StringBuilder append(Object obj) StringBuilder append(String s)	Додати аргумент до об'єкту StringBuilder. Перед тим як дані приєднуються вони перетворюються у String.
StringBuilder delete(int start, int end) StringBuilder deleteCharAt(int index)	Перший метод видаляє символи послідовність починаючи з індексу start до end-1 (включно) . Другий методи видаляє символ розташований за відповідним індексом.
StringBuilder insert(int offset, boolean b) StringBuilder insert(int offset, char c) StringBuilder insert(int offset, char[] str) StringBuilder insert(int index, char[] str, int offset, int len) StringBuilder insert(int offset, double d) StringBuilder insert(int offset, float f) StringBuilder insert(int offset, int i) StringBuilder insert(int offset, long lng) StringBuilder insert(int offset, Object obj) StringBuilder insert(int offset, String s)	Вставляє другий аргумент у послідовність. Перший цілочисельний аргумент вказує індекс перед яким дані вставляються. Перед операцією вставлення дані конвертуються у звичайний рядок String.
StringBuilder replace(int start, int end, String s) void setCharAt(int index, char c)	Заміняє зазначений символ(и)
StringBuilder reverse()	Обертає символьну послідовність ззаду наперед
String toString()	Повертає рядок типу String, що містить символьну послідовність екземпляру StringBuilder.