

Scanner Sınıfı

java.util

Class Scanner

java.lang.Object

java.util.Scanner

Kılgıladığı Arayüzler:

Closeable, AutoCloseable, Iterator<String>

Bildirimi:

```
public final class Scanner
    extends Object
        implements Iterator<String>, Closeable
```

Düzgün ifadeler kullanarak, kaynaktaki ilkel verileri ve metinleri ayıran bir metin (text) giriş aracıdır.

Bir scanner nesnesi giren veriyi parçalarına (token) ayırır. Bu ayırma işleminde parça ayırıcı öntanımlı olarak beyaz alanlardır (boşluk, tab, yeni satır –new line). Ancak, istenirse parça ayırıcı farklı tanımlanabilir. Scanner nesnesi ayırdığı parçaları ait olduğu veri tipine dönüştürür; bunu yapan farklı next metotları vardır.

Örnekler:

Aşağıdaki kodlar, klavyeden (System.in) girilen tamsayıları okur:

```
Scanner sc = new Scanner(System.in);
int i = sc.nextInt();
```

Aşağıdaki kodlar, sayılar adlı text dosyasından okuduğu sayıyı long tipine dönüştürür.

```
Scanner sc = new Scanner(new File("sayılar"));
while (sc.hasNextLong()) {
    long aLong = sc.nextLong();
}
```

Yerel sayı biçimleri

`Scanner` normal olarak işletim sisteminin kullandığı dile uyan sayı biçimini kullanır. Ama istenirse, bu biçim değiştirilebilir, yerine başka bir biçim konulabilir. Örneğin, bazı ülkelerde kesir ayracı nokta (.) iken, bazı ülkelerde ise kesir ayracı olarak virgöl (,) kullanılır. Benzer olarak binlikler ayracı, pozitif/negatif sayıyı öntakı ya da sontakı ile belirleme, okunan verinin sayı olup olmadığını belirleme gibi eylemler yapılabilir.

Scanner Sınıfının Kurucuları:

Scanner(`File` source)

Taranan dosyadan veri okumak için bir `Scanner` nesnesi yaratır.

Scanner(`File` source, `String` charsetName)

charset'i belirtilmiş olarak, taranan dosyadan veri okumak için bir `Scanner` nesnesi yaratır.

Scanner(`InputStream` source)

Taranan giriş akımından veri okumak için bir `Scanner` nesnesi yaratır.

Scanner(`InputStream` source, `String` charsetName)

charset'i belirtilmiş olarak, taranan giriş akımından veri okumak için bir `Scanner` nesnesi yaratır.

Scanner(`Path` source)

Yolu belirtilen dosyayı tarayarak veri okumak için bir `Scanner` nesnesi yaratır.

Scanner(`Path` source, `String` charsetName)

Yolu ve charset'i belirtilen dosyayı tarayarak veri okumak için bir `Scanner` nesnesi yaratır.

Scanner(`Readable` source)

Belirtilen kaynağı tarayarak veri okumak için bir `Scanner` nesnesi yaratır.

Scanner(`ReadableByteChannel` source)

Belirtilen kanalı tarayarak veri okumak için bir `Scanner` nesnesi yaratır.

Scanner(`ReadableByteChannel` source, `String` charsetName)

Belirtilen kanalı tarayarak, belirtilen charset tipinden veri okumak için bir `Scanner` nesnesi yaratır.

Scanner(`String` source)

Belirtilen `String` nesnesini tarayarak veri okumak için bir `Scanner` nesnesi yaratır.

Scanner Sınıfının Metotları:

void **close**()

Closes this scanner.

Pattern delimiter ()

Returns the `Pattern` this `Scanner` is currently using to match delimiters.

String findInLine (Pattern pattern)

Attempts to find the next occurrence of the specified pattern ignoring delimiters.

String findInLine (String pattern)

Attempts to find the next occurrence of a pattern constructed from the specified string, ignoring delimiters.

String findWithinHorizon (Pattern pattern, int horizon)

Attempts to find the next occurrence of the specified pattern.

String findWithinHorizon (String pattern, int horizon)

Attempts to find the next occurrence of a pattern constructed from the specified string, ignoring delimiters.

boolean hasNext ()

Returns true if this scanner has another token in its input.

boolean hasNext (Pattern pattern)

Returns true if the next complete token matches the specified pattern.

boolean hasNext (String pattern)

Returns true if the next token matches the pattern constructed from the specified string.

boolean hasNextBigDecimal ()

Returns true if the next token in this scanner's input can be interpreted as a `BigDecimal` using the `nextBigDecimal ()` method.

boolean hasNextBigInteger ()

Returns true if the next token in this scanner's input can be interpreted as a `BigInteger` in the default radix using the `nextBigInteger ()` method.

boolean hasNextBigInteger (int radix)

Returns true if the next token in this scanner's input can be interpreted as a `BigInteger` in the specified radix using the `nextBigInteger ()` method.

boolean hasNextBoolean ()

Returns true if the next token in this scanner's input can be interpreted as a boolean value using a case insensitive pattern created from the string "true|false".

boolean hasNextByte ()

Returns true if the next token in this scanner's input can be interpreted as a byte value in the default radix using the `nextByte()` method.

boolean **hasNextByte**(int radix)

Returns true if the next token in this scanner's input can be interpreted as a byte value in the specified radix using the `nextByte()` method.

boolean **hasNextDouble**()

Returns true if the next token in this scanner's input can be interpreted as a double value using the `nextDouble()` method.

boolean **hasNextFloat**()

Returns true if the next token in this scanner's input can be interpreted as a float value using the `nextFloat()` method.

boolean **hasNextInt**()

Returns true if the next token in this scanner's input can be interpreted as an int value in the default radix using the `nextInt()` method.

boolean **hasNextInt**(int radix)

Returns true if the next token in this scanner's input can be interpreted as an int value in the specified radix using the `nextInt()` method.

boolean **hasNextLine**()

Returns true if there is another line in the input of this scanner.

boolean **hasNextLong**()

Returns true if the next token in this scanner's input can be interpreted as a long value in the default radix using the `nextLong()` method.

boolean **hasNextLong**(int radix)

Returns true if the next token in this scanner's input can be interpreted as a long value in the specified radix using the `nextLong()` method.

boolean **hasNextShort**()

Returns true if the next token in this scanner's input can be interpreted as a short value in the default radix using the `nextShort()` method.

boolean **hasNextShort**(int radix)

Returns true if the next token in this scanner's input can be interpreted as a short value in the specified radix using the `nextShort()` method.

IOException **ioException**()

Returns the `IOException` last thrown by this `Scanner`'s underlying `Readable`.

Locale **locale**()

Returns this scanner's locale.

MatchResult **match** ()

Returns the match result of the last scanning operation performed by this scanner.

String **next** ()

Finds and returns the next complete token from this scanner.

String **next** (**Pattern** pattern)

Returns the next token if it matches the specified pattern.

String **next** (**String** pattern)

Returns the next token if it matches the pattern constructed from the specified string.

BigDecimal **nextBigDecimal** ()

Scans the next token of the input as a **BigDecimal**.

BigInteger **nextBigInteger** ()

Scans the next token of the input as a **BigInteger**.

BigInteger **nextBigInteger** (int radix)

Scans the next token of the input as a **BigInteger**.

boolean **nextBoolean** ()

Scans the next token of the input into a boolean value and returns that value.

byte **nextByte** ()

Scans the next token of the input as a **byte**.

byte **nextByte** (int radix)

Scans the next token of the input as a **byte**.

double **nextDouble** ()

Scans the next token of the input as a **double**.

float **nextFloat** ()

Scans the next token of the input as a **float**.

int **nextInt** ()

Scans the next token of the input as an **int**.

int **nextInt** (int radix)

Scans the next token of the input as an **int**.

String **nextLine** ()

Advances this scanner past the current line and returns the input that was skipped.

long **nextLong**()

Scans the next token of the input as a `long`.

long **nextLong**(int radix)

Scans the next token of the input as a `long`.

short **nextShort**()

Scans the next token of the input as a `short`.

short **nextShort**(int radix)

Scans the next token of the input as a `short`.

int **radix**()

Returns this scanner's default radix.

void **remove**()

The remove operation is not supported by this implementation of `Iterator`.

Scanner **reset**()

Resets this scanner.

Scanner **skip**(**Pattern** pattern)

Skips input that matches the specified pattern, ignoring delimiters.

Scanner **skip**(**String** pattern)

Skips input that matches a pattern constructed from the specified string.

String **toString**()

Returns the string representation of this `Scanner`.

Scanner **useDelimiter**(**Pattern** pattern)

Sets this scanner's delimiting pattern to the specified pattern.

Scanner **useDelimiter**(**String** pattern)

Sets this scanner's delimiting pattern to a pattern constructed from the specified `String`.

Scanner **useLocale**(**Locale** locale)

Sets this scanner's locale to the specified locale.

Scanner **useRadix**(int radix)

Sets this scanner's default radix to the specified radix.

java.lang.Object sınıfından kalıtsal gelen metotlar

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `wait`, `wait`, `wait`

Örnek:

Aşağıdaki program klavyeden girilen bir satırlık metni (text) okur ve ekrana yazar. Bunun için `Scanner(InputStream source)` kurucusunu kullanıyor. `InputStream` tipinden olan `source` olarak `System.in` değişkeni kullanılıyor. Bu değişken standart giriş akımıdır (klavye).

```
package scanner;

import java.util.Scanner;

public class Demo {

    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        System.out.println("Lütfen bir metin giriniz : ");
        String str = scan.nextLine();
        System.out.println(str);
    }
}
```

```
/**
Lütfen bir metin giriniz :
abc 123 def 456 ghi 78.9 jkl
abc 123 def 456 ghi 78.9 jkl
*/
```

Örnek:

Aşağıdaki program klavyeden girilen bir satırlık metni (text) okur; beyaz alanları atlayarak ekrana yazar.

```
package scanner;

import java.util.Scanner;

public class Demo {

    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        System.out.println("Lütfen bir metin giriniz : ");
        while (scan.hasNext())
        {
            String str = scan.next();
            System.out.print(str);
        }
    }
}
```

```
}

/**
Lütfen bir metin giriniz :
abc 123 def 456 ghi 78.9 jkl
abc123def456ghi78.9jkl
*/
```

Örnek:

Aşağıdaki program klavyeden girilen sayıları toplayıp ekrana yazar.

```
package scanner;

import java.util.Scanner;

public class Demo {

    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        System.out.println("Lütfen ilk tamsayıyı giriniz : ");
        int n = scan.nextInt();

        System.out.println("Lütfen ikinci tamsayıyı giriniz : ");
        int m = scan.nextInt();
        System.out.print(n + m);
    }
}
```

```
/**
Lütfen ilk tamsayıyı giriniz :
123
Lütfen ikinci tamsayıyı giriniz :
456
579
*/
```

```
package scanner;

import java.util.*;

// Sayı girdilerini okuyan Scanner örneği
public class Demo {
    public static void main(String arg[]) {

        // Klavyeden okumak için bir Scanner nesnesi yarat
        Scanner scanner = new Scanner(System.in);

        try {
```

```

5603);
        System.out.printf("int giriniz (örneğin %4d): ",
        int int_val = scanner.nextInt();
        System.out.println(" Girilen : " + int_val + "\n");

        System.out.printf("float gir : (örneğin, %5.2f): ",
        float float_val = scanner.nextFloat();
        System.out.println(" Girilen : " + float_val +
        "\n");

        System.out.printf("double giriniz (örneğin, %6.3e):
        ", 8,3456e12);
        double double_val = scanner.nextDouble();
        System.out.println("Girilen " + double_val + "\n");

    } catch (InputMismatchException e) {
        System.out.println("Uyumsuz:" + e);
    }
}
}

```

Örnek:

Aşağıdaki program bir text dosyasını okur ve ekrana yazar. Text dosyası main(String[] args) metodunun birinci parametresi olarak verilebilir. Bunu yapmak için, önce

```
Javac DosyaOku.java
```

deyimi ile DosyaOku.class bytecode'unu yaratınız. Sonra

```
Java DosyaOku örnekText
```

deyimini yazınız. Dosyanın okunup ekrana yazıldığını göreceksiniz. Tabii, okunacak örnekText dosyası ile DosyaOku.class bytecode dosyası aynı dizin içinde olmalıdır.

```

package scanner;

import java.io.File;
import java.io.FileNotFoundException;
import java.util.Scanner;

public class Demo {

    private static void readFile(String fileName) {
        try {
            File file = new File(fileName);
            Scanner scanner = new Scanner(file);
            while (scanner.hasNext()) {
                System.out.println(scanner.next());
            }
        }
    }
}

```

```

        scanner.close();
    } catch (FileNotFoundException e) {
        e.printStackTrace();
    }
}

public static void main(String[] args) {
    if (args.length != 1) {
        System.err.println("Çalıştırma: java TextScanner "
            + "okunacak dosya");
        System.exit(0);
    }
    readFile(args[0]);
}
}

```

/**

*/

Örnek:

Aşağıdaki program bir text dosyasından okuduğu double sayıları toplayıp ekrana yazıyor.

sayılar.txt dosyasını yazınız. Sonra Program 1 derlenince ortaya çıkan ScanTopla.class bytecode dosyası ile aynı dizine koyunuz.

```

Sayılar.txt:
9.4
32767
3.14159
3000000.15

```

```

package Javaio;

import java.io.FileReader;
import java.io.BufferedReader;
import java.io.IOException;
import java.util.Scanner;
import java.util.Locale;

public class ScanTopla {

    public static void main(String[] args) throws IOException {

```

```

Scanner s = null;
double sum = 0;
try {
    s = new Scanner(
        new BufferedReader(new FileReader("doubleOku.txt")));
    //s.useLocale(Locale.TR);

    while (s.hasNext()) {
        if (s.hasNextDouble()) {
            sum += s.nextDouble();
        } else {
            s.next();
        }
    }
} finally {
    s.close();
}

System.out.println(sum);
}
}
/**
 32767.0
*/

```

```

package Javaio;

import java.util.Scanner;

public class Demo {

    public static void main(String[] args) {

        String ad;
        int yaş;
        Scanner in = new Scanner(System.in);

        System.out.println("Adınız : ");
        ad = in.nextLine();

        System.out.println("Yaşınız : ");
        yaş = in.nextInt();
        in.close();

        System.out.println("Adınız : " + ad);
        System.out.println("Yaşınız : " + yaş);
    }
}
/**

```

```
Adınız :  
Demet Ceylan  
Yaşınız :  
21  
Adınız :Demet Ceylan  
Yaşınız :21  
*/
```

Java Network örneği

```
package scanner;  
  
import java.net.*;  
import java.io.IOException;  
import javax.swing.*;  
import java.awt.*;  
import java.awt.event.*;  
  
public class Demo {  
  
    public static void main(String[] args) {  
        InetAddress ia = null;  
        String host = null;  
        try {  
  
            host = JOptionPane  
                .showInputDialog("Taranacak dosya:\n  
örnek: xxx.txt");  
  
            if (host != null) {  
                ia = InetAddress.getByName(host);  
                scan(ia);  
            }  
        } catch (UnknownHostException e) {  
            System.err.println(e);  
        }  
        System.out.println("Okunan:");  
    }  
  
    public static void scan(final InetAddress remote) {  
        // menu bar değişkeni  
  
        int port = 0;  
        String hostname = remote.getHostName();  
  
        for (port = 0; port < 65536; port++) {  
            try {  
                Socket s = new Socket(remote, port);  
                System.out.println("Sunucu " + port + "  
port'u okuyor ");
```

```

        s.close();
    } catch (IOException ex) {
        System.out.println("Sunucu " + port + "
port'u okumuyor ");
    }
}
}
}

```

```

package scanner;

import java.util.*;

class Menu {
    public void display_menu() {
        System.out.println("1) Seçki 1\n2) Seçki 2\n3) Seçki 3");
        System.out.print("Seçiniz: ");
    }

    public Menu() {
        Scanner in = new Scanner(System.in);

        display_menu();
        switch (in.nextInt()) {
            case 1:
                System.out.println("Seçiminiz : Seçki 1");
                break;
            case 2:
                System.out.println("Seçiminiz : Seçki 2");
                break;
            case 3:
                System.out.println("Seçiminiz : Seçki 3");
                break;
            default:
                System.err.println("Uygunsuz Seçki");
                break;
        }
    }

    public static void main(String[] args) {
        new Menu();
    }
}

```