User Input

Tanjina Helaly

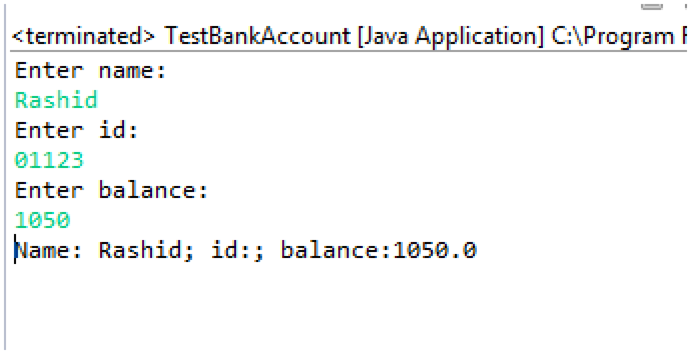
Scanner

- **Standard Java Input**
- Java assumes that you will be using a GUI for input from the user Hence there is no "simple" way to read input from a user
- Scanner class is in jdk 1.5
- In java.util package

Scanner - Example

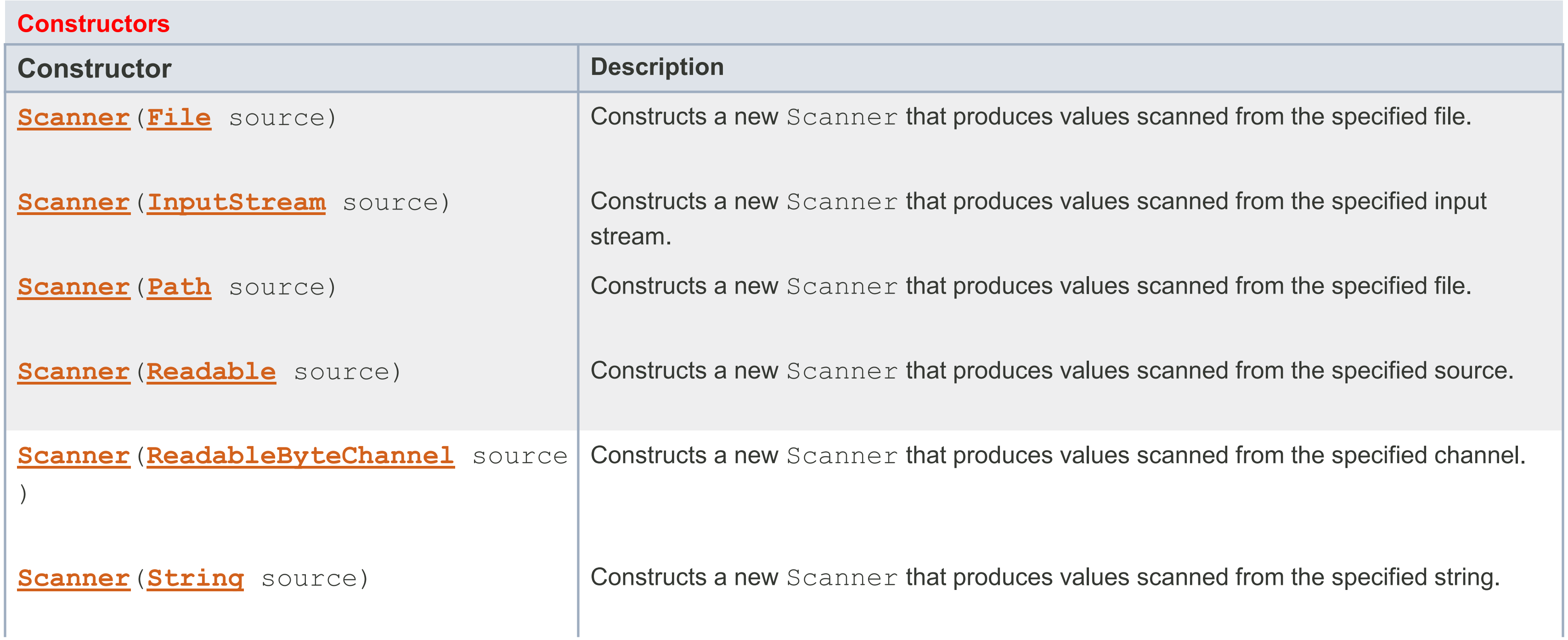
```
import java.util.Scanner;

public class TestBankAccount {
        public static void main(String[] args) {
                Scanner scan = new Scanner(System.in);
                System.out.println("Enter name:");
                String name = scan.nextLine();
                System.out.println("Enter id:");
                String id = scan.next();
                System.out.println("Enter balance:");
                double balance = scan.nextDouble();
                // Creating objects
                    BankAccount account = new BankAccount(name, "", balance);
                System.out.printf("Name: %s; id:%s; balance:%.1f", account.name, account.id, account.balance);
        }
}
```

Output



```
<terminated> TestBankAccount [Java Application] C:\Program F
Enter name:
Rashid
Enter id:
01123
Enter balance:
1050
Name: Rashid; id:; balance:1050.0
```

Scanner Constructors



Constructors

Constructor	Description
`Scanner(File source)`	Constructs a new `Scanner` that produces values scanned from the specified file.
`Scanner(InputStream source)`	Constructs a new `Scanner` that produces values scanned from the specified input stream.
`Scanner(Path source)`	Constructs a new `Scanner` that produces values scanned from the specified file.
`Scanner(Readable source)`	Constructs a new `Scanner` that produces values scanned from the specified source.
`Scanner(ReadableByteChannel source)`	Constructs a new `Scanner` that produces values scanned from the specified channel.
`Scanner(String source)`	Constructs a new `Scanner` that produces values scanned from the specified string.

SCANNER METHODS

Methods

Modifier and Type	Method and Description
`void`	`close()` Closes this scanner.
`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`	`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.
`String`	`next()` Finds and returns the next complete token from this scanner.
`String`	`next(String pattern)` Returns the next token if it matches the pattern constructed from the specified string.
`BigInteger`	`nextBigInteger()` Scans the next token of the input as a `BigInteger`.

Scanner Methods

Methods

Modifier and Type	Method and Description
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.
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.
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.
short	nextShort() Scans the next token of the input as a short.
Scanner	reset() Resets this scanner.

JOptionPane

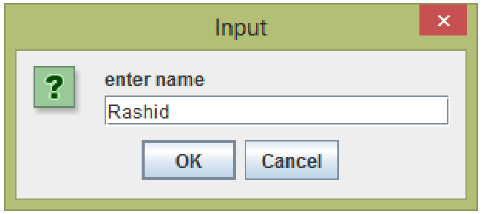
- JOptionPane makes it easy to pop up a standard dialog box that prompts users for a value or informs them of something.

Method Name	Description
showConfirmDialog	Asks a confirming question, like yes/no/cancel.
showInputDialog	Prompt for some input.
showMessageDialog	Tell the user about something that has happened.
showOptionDialog	The Grand Unification of the above three.

- https://docs.oracle.com/javase/7/docs/api/javax/swing/JOptionPane.html

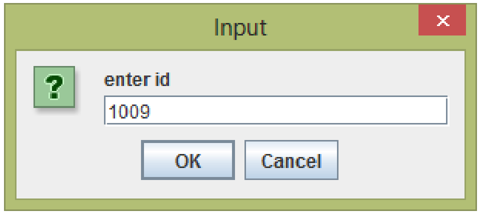
JOptionPane - Example

```
import javax.swing.JOptionPane;

public class TestBankAccount {
    public static void main(String[] args) {
        String name = JOptionPane.showInputDialog(null, "enter name");
        String id = JOptionPane.showInputDialog(null, "enter id");
        String inpBalance = JOptionPane.showInputDialog(null, "enter balance");
        double balance = Double.parseDouble(inpBalance);

        // Creating objects
           BankAccount account = new BankAccount(name, "", balance);
        System.out.printf("Name: %s; id:%s; balance:%.1f", account.name, account.id, account.balance);
    }
}
```



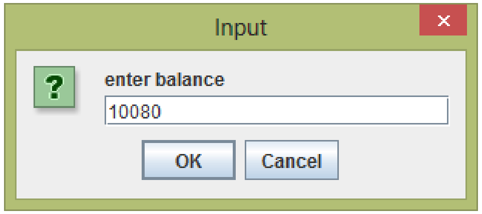
enter name

Rashid



enter id

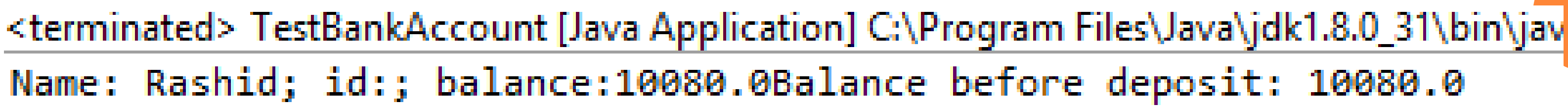
1009



enter balance

10080

Output



```
<terminated> TestBankAccount [Java Application] C:\Program Files\Java\jdk1.8.0_31\bin\jav
Name: Rashid; id:; balance:10080.0Balance before deposit: 10080.0
```