In one scenario, a program dialogs with the user. The program prompts the user for input by using, say, methods

System.out.print(…) and System.out.println(…)

The user then types information on the keyboard, which the program reads. We show you how to do this using class java.util.Scanner. We then summarize other methods of class Scanner that may be useful to you.

**System.in and Scanner**

The line below creates variable sc so that it can be used to read from the keyboard, which is given by File System.in:

Scanner sc = **new** Scanner(System.in);

One can then call method sc.nextLine(); to read the next line of input from the keyboard and return a String. The method call waits until the reader clicks the return/enter key and then returns the String of characters that were typed in.

/\*\* Add the integers the user types (one at a time, with

\* a return after each) and show the running sum.

\* Stop when the user types STOP in caps. \*/

public static void main(String[] args) {

Scanner sc= new Scanner(System.in);

int sum= 0;

// invariant: sum is the sum of integers the user typed

while (true) {

System.out.println("Type integer, or STOP to end");

String s= sc.nextLine().trim();

if (s.equals("STOP")) {

System.out.println("Goodbye");

return;

}

try {

int k= Integer.parseInt(s);

sum= sum + k;

System.out.println("The running sum is: " + sum);

} catch (NumberFormatException n) {

System.out.println("That wasn't an integer.");

}

}

Method main, to the right, illustrates this, showing all the points that such a method may have to deal with. Read its specification to see what it does.

1. The while-loop has condition true; its repetend will execute a return statement to terminate the program.

2. In the repetend, first, a println statement prompts the user for input.

3. Note how leading and trailing blanks are trimmed from the string that is read in.

4. The call to parseInt is within a try-block because the user may have made a mistake in typing an integer. If no mistake is made, the integer is added to variable sum and the running sum is printed. If a mistake is made, the catch-block prints a message to the user.

**Other methods of class Scanner**

Class Scanner is a powerful tool, allowing any input String —not just characters from the keyboard— to be parsed in complicated ways. Read its specification for more information.

A user might type several things on a line before hitting the return/enter key, separating them with blanks or tabs. Class Scanner has hasNext…() methods to determine what is next in the input String and next…() methods to return the next thing. When the input is coming from the keyboard, all these methods wait until the user has hit the return/enter key.

hasNextBoolean() nextBoolean()

hasNextDouble() nextDouble()

hasNextInt() nextInt()

hasNextLine() nextLine()