/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Compilation: javac StdOut.java

\* Execution: java StdOut

\* Dependencies: none

\*

\* Writes data of various types to standard output.

\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

import java.io.OutputStreamWriter;

import java.io.PrintWriter;

import java.io.UnsupportedEncodingException;

import java.util.Locale;

/\*\*

\* This class provides methods for printing strings and numbers to standard output.

\* <p>

\* <b>Getting started.</b>

\* To use this class, you must have {@code StdOut.class} in your

\* Java classpath. If you used our autoinstaller, you should be all set.

\* Otherwise, download

\* <a href = "http://introcs.cs.princeton.edu/java/stdlib/StdOut.java">StdOut.java</a>

\* and put a copy in your working directory.

\* <p>

\* Here is an example program that uses {@code StdOut}:

\* <pre>

\* public class TestStdOut {

\* public static void main(String[] args) {

\* int a = 17;

\* int b = 23;

\* int sum = a + b;

\* StdOut.println("Hello, World");

\* StdOut.printf("%d + %d = %d\n", a, b, sum);

\* }

\* }

\* </pre>

\* <p>

\* <b>Differences with System.out.</b>

\* The behavior of {@code StdOut} is similar to that of {@link System#out},

\* but there are a few subtle differences:

\* <ul>

\* <li> {@code StdOut} coerces the character-set encoding to UTF-8,

\* which is a standard character encoding for Unicode.

\* <li> {@code StdOut} coerces the locale to {@link Locale#US},

\* for consistency with {@link StdIn}, {@link Double#parseDouble(String)},

\* and floating-point literals.

\* <li> {@code StdOut} <em>flushes</em> standard output after each call to

\* {@code print()} so that text will appear immediately in the terminal.

\* </ul>

\* <p>

\* <b>Reference.</b>

\* For additional documentation,

\* see <a href="http://introcs.cs.princeton.edu/15inout">Section 1.5</a> of

\* <em>Computer Science: An Interdisciplinary Approach</em>

\* by Robert Sedgewick and Kevin Wayne.

\*

\* @author Robert Sedgewick

\* @author Kevin Wayne

\*/

public final class StdOut {

// force Unicode UTF-8 encoding; otherwise it's system dependent

private static final String CHARSET\_NAME = "UTF-8";

// assume language = English, country = US for consistency with StdIn

private static final Locale LOCALE = Locale.US;

// send output here

private static PrintWriter out;

// this is called before invoking any methods

static {

try {

out = new PrintWriter(new OutputStreamWriter(System.out, CHARSET\_NAME), true);

}

catch (UnsupportedEncodingException e) {

System.out.println(e);

}

}

// don't instantiate

private StdOut() { }

/\*\*

\* Closes standard output.

\*/

public static void close() {

out.close();

}

/\*\*

\* Terminates the current line by printing the line-separator string.

\*/

public static void println() {

out.println();

}

/\*\*

\* Prints an object to this output stream and then terminates the line.

\*

\* @param x the object to print

\*/

public static void println(Object x) {

out.println(x);

}

/\*\*

\* Prints a boolean to standard output and then terminates the line.

\*

\* @param x the boolean to print

\*/

public static void println(boolean x) {

out.println(x);

}

/\*\*

\* Prints a character to standard output and then terminates the line.

\*

\* @param x the character to print

\*/

public static void println(char x) {

out.println(x);

}

/\*\*

\* Prints a double to standard output and then terminates the line.

\*

\* @param x the double to print

\*/

public static void println(double x) {

out.println(x);

}

/\*\*

\* Prints an integer to standard output and then terminates the line.

\*

\* @param x the integer to print

\*/

public static void println(float x) {

out.println(x);

}

/\*\*

\* Prints an integer to standard output and then terminates the line.

\*

\* @param x the integer to print

\*/

public static void println(int x) {

out.println(x);

}

/\*\*

\* Prints a long to standard output and then terminates the line.

\*

\* @param x the long to print

\*/

public static void println(long x) {

out.println(x);

}

/\*\*

\* Prints a short integer to standard output and then terminates the line.

\*

\* @param x the short to print

\*/

public static void println(short x) {

out.println(x);

}

/\*\*

\* Prints a byte to standard output and then terminates the line.

\* <p>

\* To write binary data, see {@link BinaryStdOut}.

\*

\* @param x the byte to print

\*/

public static void println(byte x) {

out.println(x);

}

/\*\*

\* Flushes standard output.

\*/

public static void print() {

out.flush();

}

/\*\*

\* Prints an object to standard output and flushes standard output.

\*

\* @param x the object to print

\*/

public static void print(Object x) {

out.print(x);

out.flush();

}

/\*\*

\* Prints a boolean to standard output and flushes standard output.

\*

\* @param x the boolean to print

\*/

public static void print(boolean x) {

out.print(x);

out.flush();

}

/\*\*

\* Prints a character to standard output and flushes standard output.

\*

\* @param x the character to print

\*/

public static void print(char x) {

out.print(x);

out.flush();

}

/\*\*

\* Prints a double to standard output and flushes standard output.

\*

\* @param x the double to print

\*/

public static void print(double x) {

out.print(x);

out.flush();

}

/\*\*

\* Prints a float to standard output and flushes standard output.

\*

\* @param x the float to print

\*/

public static void print(float x) {

out.print(x);

out.flush();

}

/\*\*

\* Prints an integer to standard output and flushes standard output.

\*

\* @param x the integer to print

\*/

public static void print(int x) {

out.print(x);

out.flush();

}

/\*\*

\* Prints a long integer to standard output and flushes standard output.

\*

\* @param x the long integer to print

\*/

public static void print(long x) {

out.print(x);

out.flush();

}

/\*\*

\* Prints a short integer to standard output and flushes standard output.

\*

\* @param x the short integer to print

\*/

public static void print(short x) {

out.print(x);

out.flush();

}

/\*\*

\* Prints a byte to standard output and flushes standard output.

\*

\* @param x the byte to print

\*/

public static void print(byte x) {

out.print(x);

out.flush();

}

/\*\*

\* Prints a formatted string to standard output, using the specified format

\* string and arguments, and then flushes standard output.

\*

\*

\* @param format the <a href = "http://docs.oracle.com/javase/7/docs/api/java/util/Formatter.html#syntax">format string</a>

\* @param args the arguments accompanying the format string

\*/

public static void printf(String format, Object... args) {

out.printf(LOCALE, format, args);

out.flush();

}

/\*\*

\* Prints a formatted string to standard output, using the locale and

\* the specified format string and arguments; then flushes standard output.

\*

\* @param locale the locale

\* @param format the <a href = "http://docs.oracle.com/javase/7/docs/api/java/util/Formatter.html#syntax">format string</a>

\* @param args the arguments accompanying the format string

\*/

public static void printf(Locale locale, String format, Object... args) {

out.printf(locale, format, args);

out.flush();

}

/\*\*

\* Unit tests some of the methods in {@code StdOut}.

\*

\* @param args the command-line arguments

\*/

public static void main(String[] args) {

// write to stdout

StdOut.println("Test");

StdOut.println(17);

StdOut.println(true);

StdOut.printf("%.6f\n", 1.0/7.0);

}

}