# Task 7.1

## Sphinx.java

**import** java.util.Scanner;

**public** **class** Sphnix {

**public** **static** **int** unknown(**int** x, **int** y) {

**if** (x < y) {

**return** x;

} **else** {

**return** y;

}

}

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

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

System.***out***.print("x: ");

**int** x = input.nextInt();

System.***out***.print("y: ");

**int** y = input.nextInt();

System.***out***.println();

System.***out***.println("unknown = " + *unknown*(x, y));

System.***out***.println();

}

}

# Task 7.2

## BadFormatting.java

/\*\*

\* This code is an example code using the Stack method.

\* **@author** 991112

\*

\*/

**public** **class** BadFormatting {

**private** **static** **int** *stackSize* = 5;

**private** **static** **int** *topOfStack* = -1;

**private** **static** **int**[] *stack* = **new** **int**[*stackSize*];

**private** **static** **boolean** *errorFree* = **true**;

/\*\*

\* An empty constructor for the class badFormatting.

\*/

**public** BadFormatting() {

}

/\*\*

\* If the stack is empty, the top of the stack is returned as being -1.

\* **@return** topOfStack

\*/

**public** **static** **boolean** isEmpty() {

**return** *topOfStack* == -1;

}

/\*\*

\* If the stack is full, the topOfStack is equal to the stackSize -1.

\* **@return** the top of the stack, now -1 in size.

\*/

**public** **static** **boolean** isFull() {

**return** *topOfStack* == *stackSize* - 1;

}

/\*\*

\* A method to check that the stack, if empty, and error free sets

\* -1 as the top of the stack.

\*/

**public** **static** **void** empty() {

*errorFree* = **true**;

*topOfStack* = -1;

}

/\*\*

\* This method, if error free, returns the top of stack if the stack

\* has values. Else it returns 0.

\* **@return** element at top of stack (if error free), else 0

\*/

**public** **static** **int** top() {

*errorFree* = !(*isEmpty*()) & *errorFree*;

**if** (*errorFree*) {

**return** *stack*[*topOfStack*];

} **else** {

**return** 0;

}

}

/\*\*

\* This method, if error free, and if the stack isn't full

\* adds one space to the top of stack and inserts a value.

\* **@param** value the value to be added to the stack

\*/

**public** **static** **void** push(**int** value) {

*errorFree* = !(*isFull*()) & *errorFree*;

**if** (*errorFree*) {

*topOfStack* = *topOfStack* + 1;

*stack*[*topOfStack*] = value;

}

}

/\*\*

\* The pop method deletes an element at the top of the stack.

\*/

**public** **static** **void** pop() {

*errorFree* = !(*isEmpty*()) & *errorFree*;

**if** (*errorFree*) {

*topOfStack* = *topOfStack* - 1;

}

}

/\*\*

\* This main method tests the Stack method.

\* **@param** args main args

\*/

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

**int** test;

System.***out***.println(" --- Begin Experiment 1 ---");

System.***out***.println("Build up a stack of one entry:");

*empty*();

*push*(1);

System.***out***.println(" push 1");

System.***out***.println("Inspect the stack:");

test = *top*();

System.***out***.println(" top: " + test);

System.***out***.println("Make the stack empty:");

*pop*();

System.***out***.println(" pop");

System.***out***.println("Test how 'top' works on the empty stack:");

test = *top*();

**if** (*errorFree*) {

System.***out***.println(" top: " + test);

} **else** {

System.***out***.println(" An error has occured.");

}

System.***out***.println(" --- End Experiment 1 ---");

*empty*();

System.***out***.println(" --- Empty ---");

System.***out***.println(" --- Begin Experiment 2 ---");

System.***out***.println("Build up a stack of five entries:");

*push*(1);

System.***out***.println(" push 1");

*push*(2);

System.***out***.println(" push 2");

*push*(3);

System.***out***.println(" push 3");

*push*(4);

System.***out***.println(" push 4");

*push*(5);

System.***out***.println(" push 5");

System.***out***.println("Push another entry and check if "

+ "'out of memory'-protection works:");

*push*(6);

**if** (*errorFree*) {

System.***out***.println(" push 6");

} **else** {

System.***out***.println(" An error has occured.");

}

System.***out***.println(" --- End Experiment 2 ---");

*empty*();

System.***out***.println(" --- Empty ---");

System.***out***.println(" --- Begin Experiment 3 ---");

System.***out***.println("Build up a stack of three entries:");

*push*(1);

System.***out***.println(" push 1");

*push*(2);

System.***out***.println(" push 2");

*push*(3);

System.***out***.println(" push 3");

System.***out***.println("Take these three entries away.");

**while** (!*isEmpty*()) {

test = *top*();

System.***out***.println(" top: " + test);

*pop*();

System.***out***.println(" pop");

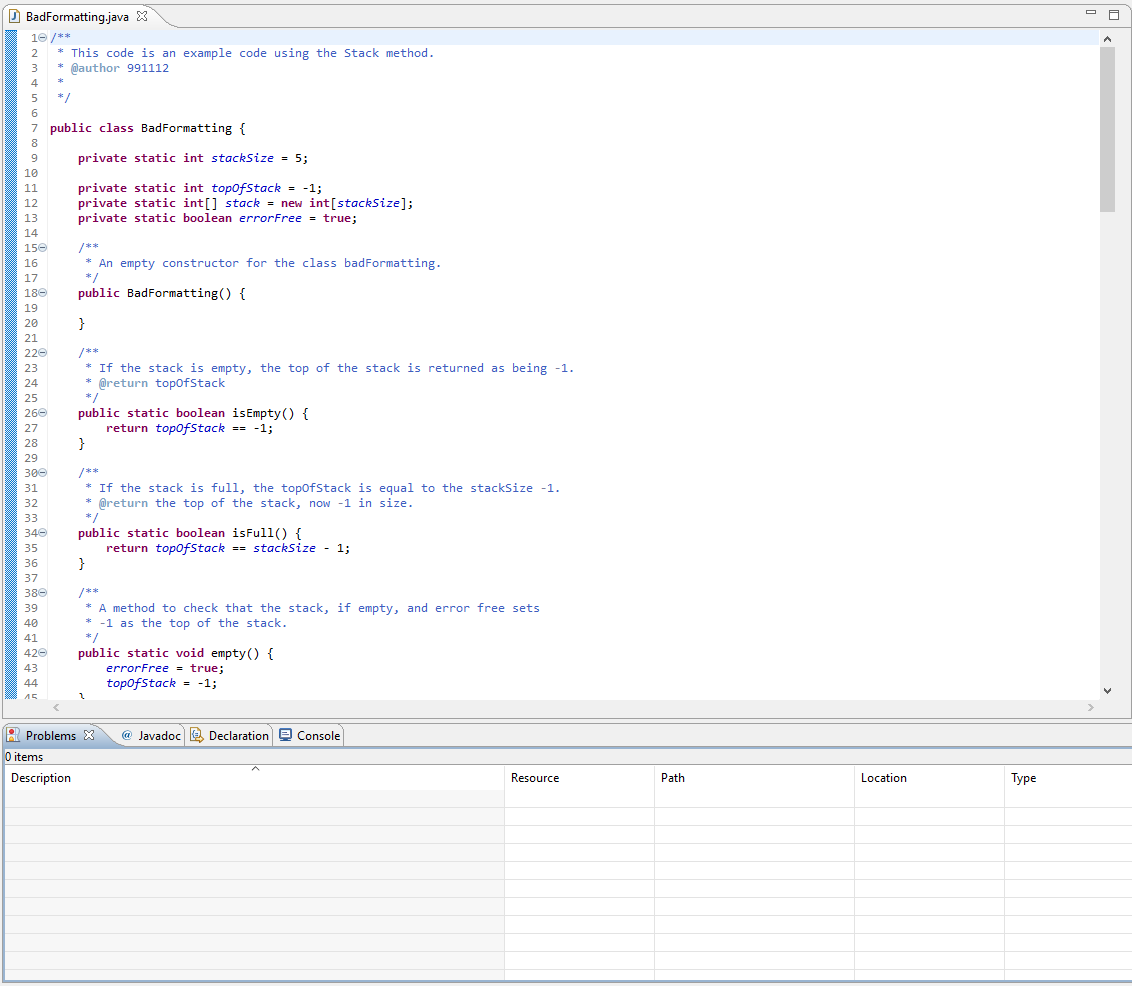
}

System.***out***.println(" --- End Experiment 3 ---");

}

}

## Check Style Passed



No Checkstyle problems found