HELLO EVERYONE
imgflip.com

How was the homework?

And how are you?

whihle vs. do-while

do { <<body>> } while(<<condition>>);

while(<<condition>>) { <<body>> }

?? Differences ??

whihle vs. do-while

do { <<body>> } while(<<condition>>);

while(<<condition>>) { <<body>> }

?? Differences ??

Do-while: <<body>> get’s executed at least once.
While: Condition gets tested BEFORE <<body>> is executed

Functions

What is the “fun” in functions?

Adding two numbers read from the console

The code is tiny, I know :S

The way you would do it now :)

```
package funktionen;

import java.io.BufferedReader;

import java.io.InputStreamReader;

public class Main {   public static void main(String[] args) {

       BufferedReader console = new BufferedReader(new InputStreamReader(System.in));

       String line = null;

       int sum = 0;

       for (int i = 0; i < 2; i++) {

           try {

                    System.out.print("Input a number: ");

                    line = console.readLine();

                    sum += Integer.parseInt(line);

                } catch (Exception e) {

                    i--;

                }

            }

            System.out.printf("Sum is %s%n", sum);

        }

      }
```

Lets make a function out of that :D

```java
package funktionen;

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

public class Main2 {

   public static void main(String[] args) {

       System.out.printf("Sum is %s%n", readInt() + readInt());

   }

   public static int readInt() {

       BufferedReader console = new BufferedReader(new InputStreamReader(System.in));

       boolean valid = false;

           int value = 0;

           do {

               try {

                   System.out.print("Input a number: ");

                   value = Integer.parseInt(console.readLine());

                   valid = true;

               } catch (NumberFormatException e) {

                   System.out.printf("Not a valid number. Reason: \"%s\"%n", e);

               } catch (IOException e) {

                   System.out.println("Unable to find the console....");

               }

           } while (!valid);

           return value;

       }

   }
```

Visibility - Private vs. public (find the problem)

```java
package funktionen;

public class A {

   public static void main(String[] args) {

       A.doPrivate();

       A.doPublic();

   }

   private static void doPrivate() {    }

   public static void doPublic() {    }

}
```

```java
package funktionen;

public class B {

   public static void main(String[] args) {

       A.doPrivate();

       A.doPublic();

   }

}
```

Writing Algorithms

If you sit down in front of a computer and try to write a program to solve a problem, you should be trying to do this:

Writing Algorithms

If you sit down in front of a computer and try to write a program to solve a problem, you should be trying to do this:

1. Understand/Analyse the problem
   Are you making a pie? Are you making a game?

Writing Algorithms

If you sit down in front of a computer and try to write a program to solve a problem, you should be trying to do this:

1. Understand/Analyse the problem
   Are you making a pie? Are you making a game?
2. Design a program
   What are the required steps?

Writing Algorithms

If you sit down in front of a computer and try to write a program to solve a problem, you should be trying to do this:

1. Understand/Analyse the problem
   Are you making a pie? Are you making a game?
2. Design a program
   What are the required steps?
3. Code the program
   Does it work? Is each step correct? Necessary?

Writing Algorithms

If you sit down in front of a computer and try to write a program to solve a problem, you should be trying to do this:

1. Understand/Analyse the problem
   Are you making a pie? Are you making a game?
2. Design a program
   What are the required steps?
3. Code the program
   Does it work? Is each step correct? Necessary?
4. Test the program (is the solution accurate)
   Will it always lead to a solution

What is an Algorithm?

A step-by-step set of instructions for solving a problem in a limited number of steps. The instructions for each step are exact and precise and can be carried out by a computer.

In simple terms, it is possible to say that an algorithm is a sequence of steps which allow to solve a certain task.

Notice

- Notice the term limited (finite). Algorithmus should lead to an eventual solution. **Or do they?**

- Step by step process. Each step should do one logical action

Examples Of Algorithms

Problem: To find the sum of two numbers

Algorithm:

Examples Of Algorithms

Problem: To find the sum of two numbers

Algorithm:

$$\left.\begin{array}{l}\text{1. add the two numbers}\\ \text{2. write down the answer}\end{array}\right\}\begin{array}{c}\text{STEPPING}\\ \text{OUT}\end{array}$$

Algorithms In Programming

In programming, algorithm are the set of well defined instruction in sequence to solve a program. An algorithm should always have a clear stopping point.

Algorithms In Programming

Exercise: Write an algorithm to find and display the sum of two input numbers :

Algorithms In Programming

Step 1: Start
Step 2: Declare variables num1, num2 and sum.

Algorithms In Programming

```
Step 1: Start
Step 2: Declare variables num1, num2 and sum.
Step 3: Read values num1 and num2.
```

Algorithms In Programming

```
Step 1: Start
Step 2: Declare variables num1, num2 and sum.
Step 3: Read values num1 and num2.
Step 4: Add num1 and num2 and assign the result
        to sum.
                sum←num1+num2
```

Algorithms In Programming

```
Step 1: Start
Step 2: Declare variables num1, num2 and sum.
Step 3: Read values num1 and num2.
Step 4: Add num1 and num2 and assign the result
          to sum.
                    $sum \leftarrow num1+num2$
Step 5: Display sum
Step 6: Stop
```

EXERCISE:

Problem 1. Find the average of four numbers

Problem 2. Write an algorithm to find the largest among three different numbers entered by user.

Problem 3. Change the time in seconds to minutes.

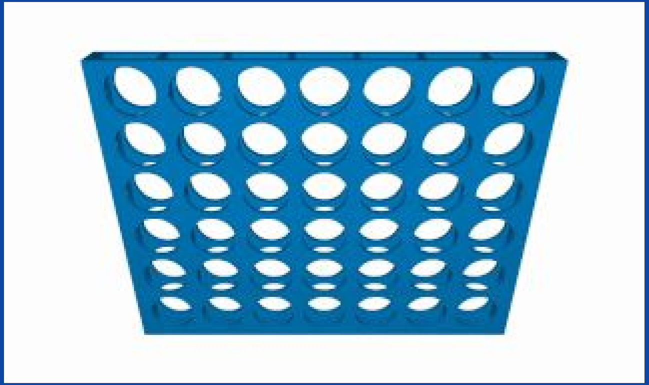
Break

https://www.pinterest.com/pin/118430665176999497/

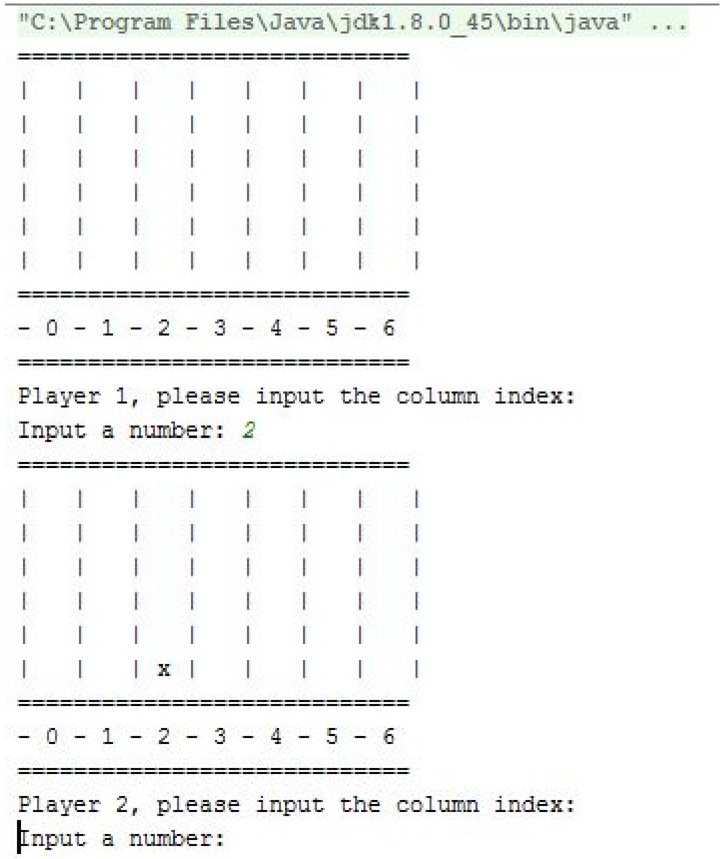
Project time

4 in a row

What’s 4 in a row?



How does the result look like?

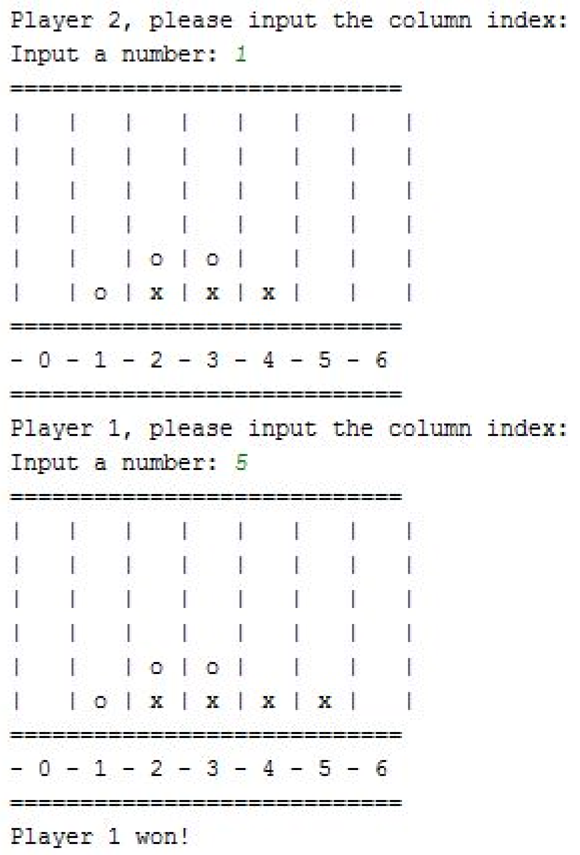


1)

```
"C:\Program Files\Java\jdk1.8.0_45\bin\java" ...
==============================
|   |   |   |   |   |   |   |
|   |   |   |   |   |   |   |
|   |   |   |   |   |   |   |
|   |   |   |   |   |   |   |
|   |   |   |   |   |   |   |
|   |   |   |   |   |   |   |
==============================
- 0 - 1 - 2 - 3 - 4 - 5 - 6
==============================
Player 1, please input the column index:
Input a number: 2
```

2)

```
==============================
|   |   |   |   |   |   |   |
|   |   |   |   |   |   |   |
|   |   |   |   |   |   |   |
|   |   |   |   |   |   |   |
|   |   |   |   |   |   |   |
|   |   | x |   |   |   |   |
==============================
- 0 - 1 - 2 - 3 - 4 - 5 - 6
==============================
Player 2, please input the column index:
Input a number:
```

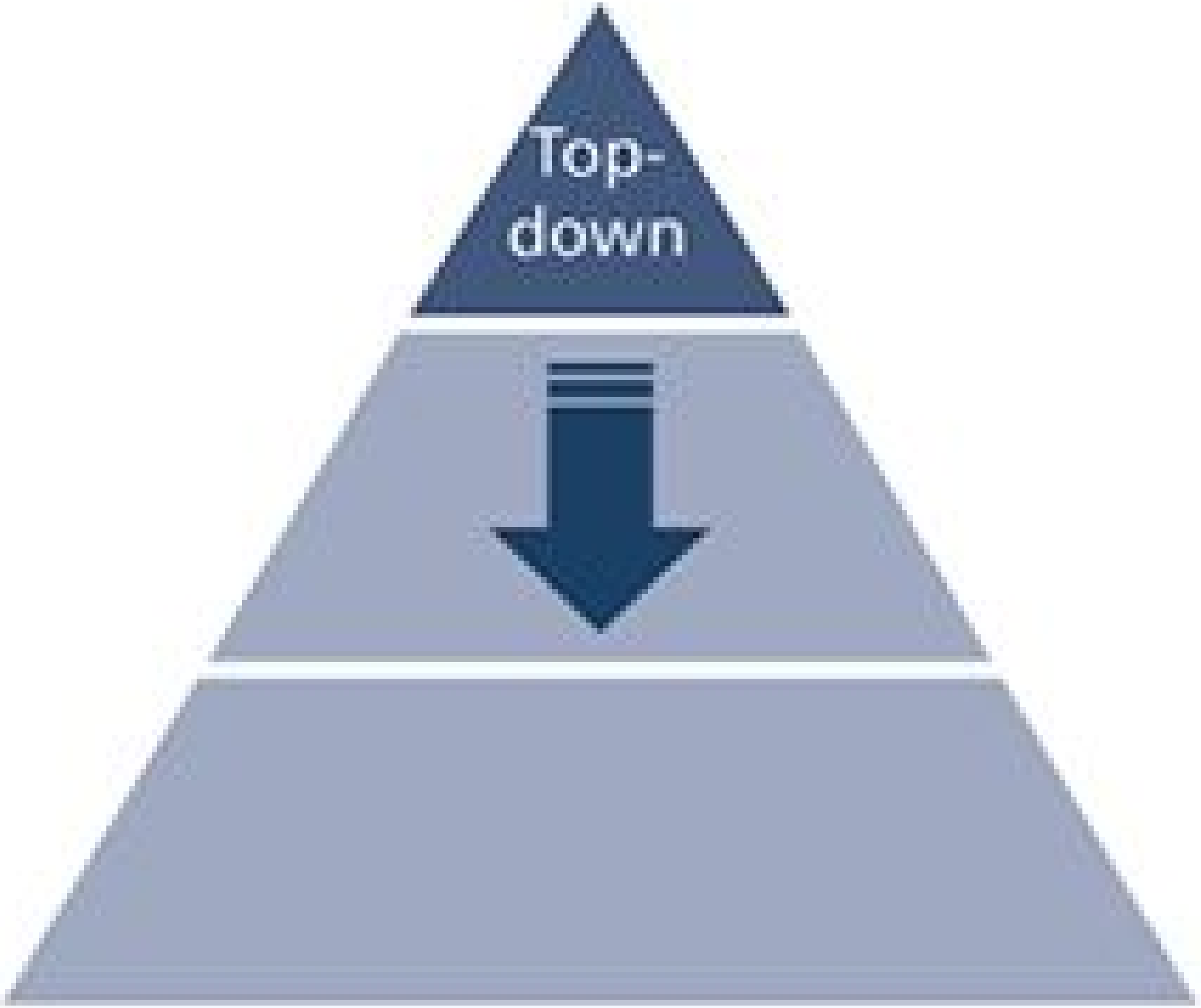


[n-1])

```
Player 2, please input the column index:
Input a number: 1
==============================
|   |   |   |   |   |   |   |
|   |   |   |   |   |   |   |
|   |   |   |   |   |   |   |
|   |   |   |   |   |   |   |
|   |   | o | o |   |   |   |
|   | o | x | x | x |   |   |
==============================
- 0 - 1 - 2 - 3 - 4 - 5 - 6
==============================
Player 1, please input the column index:
Input a number: 5
```

n)

```
==============================
|   |   |   |   |   |   |   |
|   |   |   |   |   |   |   |
|   |   |   |   |   |   |   |
|   |   |   |   |   |   |   |
|   |   | o | o |   |   |   |
|   | o | x | x | x | x |   |
==============================
- 0 - 1 - 2 - 3 - 4 - 5 - 6
==============================
Player 1 won!
```


Top-
down

Going LOOPY !

Problem: to wash a car

Algorithm:

```
1. Repeat
2. wash with warm soapy water
3. UNTIL the whole car is clean
```

compare it to STEPPING OUT!

4 in a row - Algorithm

Step 1: Start

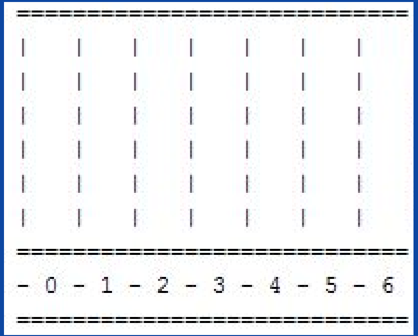
Step 2: Declare variables:
player1, player2, currentPlayer

Step 3: Set currentPlayer to be player2.

4 in a row - Algorithm

Step 4: Declare and initialize variable field
(playground: 6 rows by 7 columns)

Step 5: Draw the playground



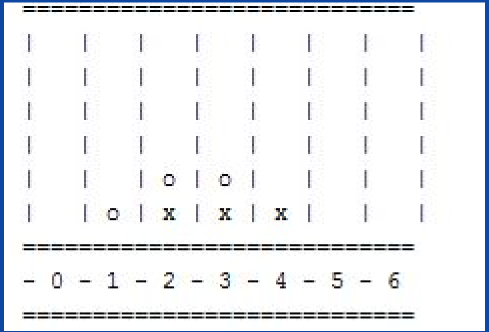
```
============================
|   |   |   |   |   |   |
|   |   |   |   |   |   |
|   |   |   |   |   |   |
|   |   |   |   |   |   |
|   |   |   |   |   |   |
|   |   |   |   |   |   |
============================
- 0 - 1 - 2 - 3 - 4 - 5 - 6
============================
```

Step 6: Ask current player to enter
a column number

4 in a row - Algorithm

Step 7: Redraw the updated playground after each new input from a player

Occupied fields are:
x for the player 1,
0 for the player 2



```
=============================
|   |   |   |   |   |   |   |
|   |   |   |   |   |   |   |
|   |   |   |   |   |   |   |
|   |   |   |   |   |   |   |
|   |   | o | o |   |   |   |
|   | o | x | x | x |   |   |
=============================
- 0 - 1 - 2 - 3 - 4 - 5 - 6
=============================
```

4 in a row - Algorithm

Step 8: Process input from a player

The expected output must be one of the following strings:

‘Player 1 won’ if the game ended and player 1 won
‘Player 2 won’ if the game ended and player 2 won
‘Draw’ if the game ended with no winner
‘Input a number’ if the game did not ended yet

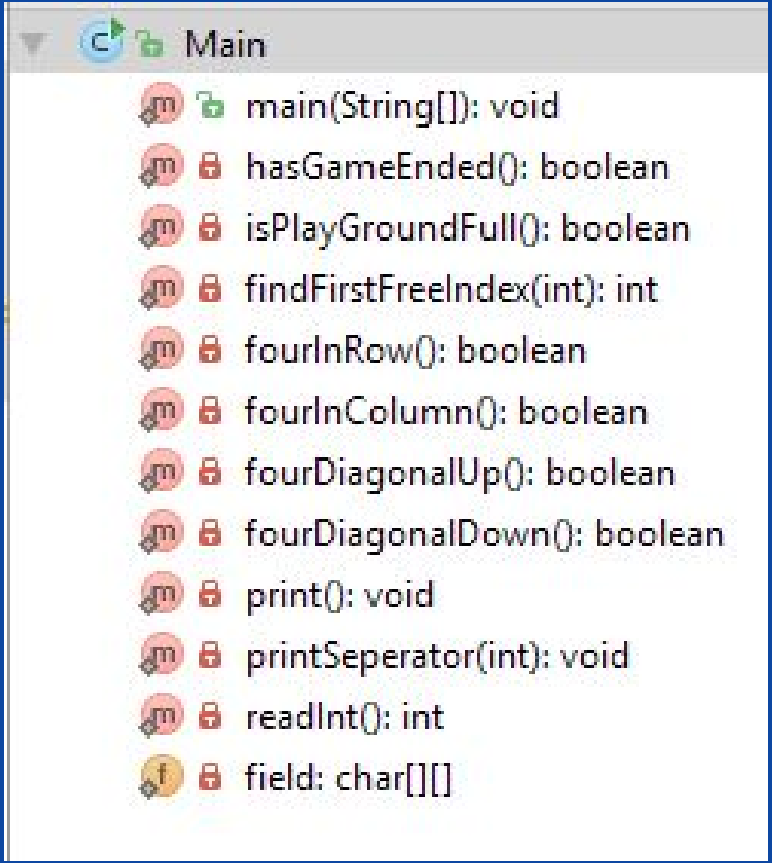
4 in a row - Algorithm

Step 9: While the game is still running continue with switching between the two players and each time you change the player ask the current player to enter a column number (afterwards proceed again from the Step 8).

4 in a row - Code

Code the program ....

4 in a row - Cheatsheet



```
Main
    main(String[]): void
    hasGameEnded(): boolean
    isPlayGroundFull(): boolean
    findFirstFreeIndex(int): int
    fourInRow(): boolean
    fourInColumn(): boolean
    fourDiagonalUp(): boolean
    fourDiagonalDown(): boolean
    print(): void
    printSeperator(int): void
    readInt(): int
    field: char[][]
```

Solution of Problem 2

```
Step 1: Start
Step 2: Declare variables a,b and c.
Step 3: Read variables a,b and c.
Step 4: If a>b
            If a>c
                Display a is the largest number.
            Else
                Display c is the largest number.
        Else
            If b>c
                Display b is the largest number.
            Else
                Display c is the greatest number.
Step 5: Stop
```