

# Java is easy

# Presentation

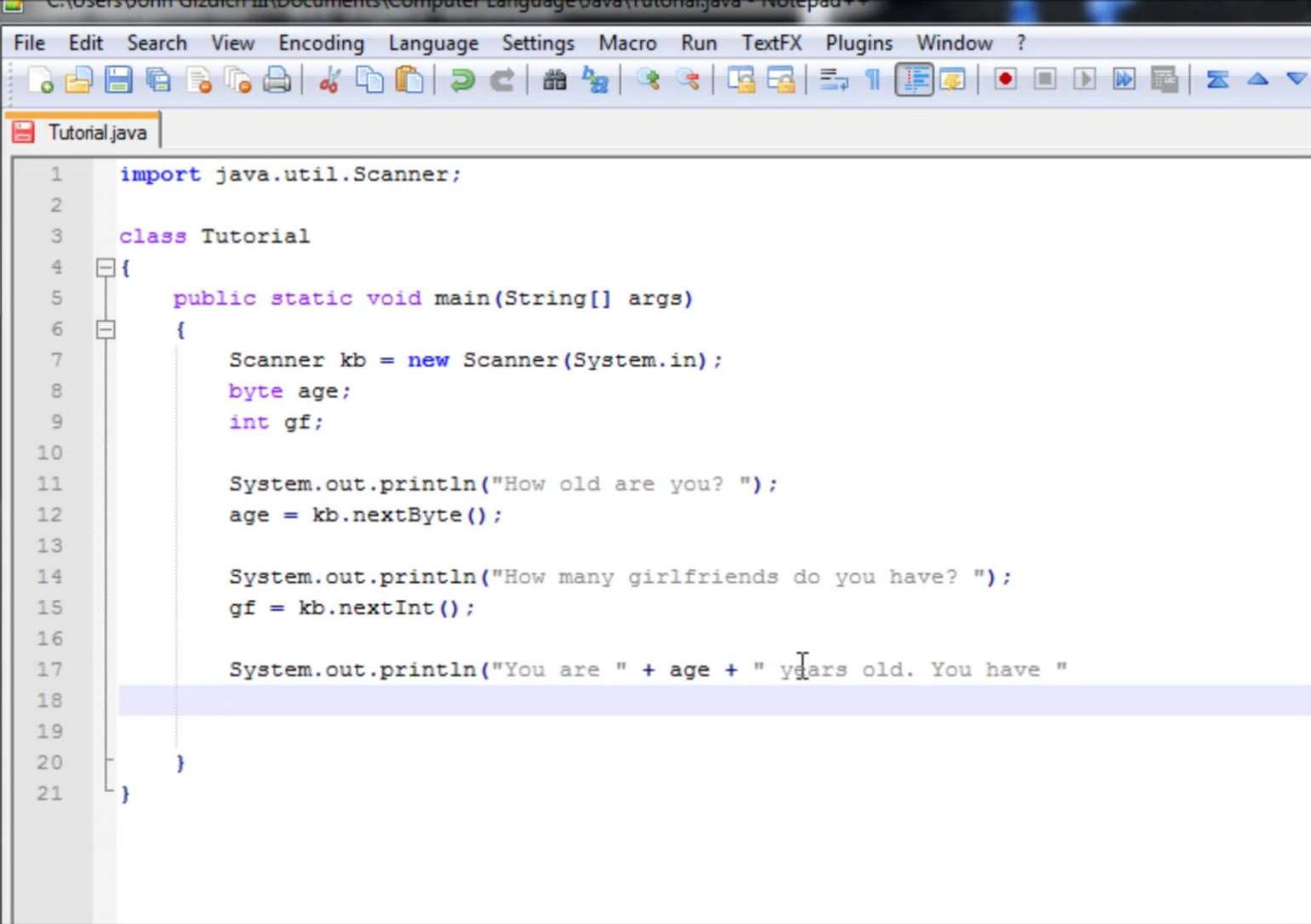


# Today's Agenda

1. Knowing the *class* **Scanner**: the **most** modern way to make **IO**(input/output) in **Java**
2. Finding some **FIRST class** literature to put big and solid grounds for the future
3. Working with **2D** arrays in order to display, organise and calculate **our common world** better
4. Career chances and project work: the **key** for **success**
5. The **Homework!**-> the game *4 in a row* (Old but **Gold**)

# 1. Knowing the *class* **Scanner**: the **most** modern way to make **IO**(input/output) in **Java**

Simple  
example:



The screenshot shows a Java IDE window titled 'Tutorial.java'. The code is as follows:

```
1  import java.util.Scanner;
2
3  class Tutorial
4  {
5      public static void main(String[] args)
6      {
7          Scanner kb = new Scanner(System.in);
8          byte age;
9          int gf;
10
11          System.out.println("How old are you? ");
12          age = kb.nextByte();
13
14          System.out.println("How many girlfriends do you have? ");
15          gf = kb.nextInt();
16
17          System.out.println("You are " + age + " years old. You have "
18
19
20      }
21 }
```

```

import java.util.InputMismatchException;
import java.util.Scanner;

public class Except {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        boolean bError = true;
        int n1 = 0, n2 = 0, nQuotient = 0;

        do {
            try {
                System.out.println("Enter first num: ");
                n1 = input.nextInt();
                System.out.println("Enter second num: ");
                n2 = input.nextInt();
                nQuotient = n1/n2;
                bError = false;
            }
            catch (Exception e) {
                System.out.println("Error!");
            }
        } while (bError);

        System.out.printf("%d/%d = %d",n1,n2, nQuotient);
    }
}

```

Full example with  
try/catch block

## 2. Finding some **FIRST** class literature to put big and solid grounds for the future

The screenshot shows a web browser window with multiple tabs. The active tab is 'accu.org', displaying the 'Book Reviews' section. The page features the ACCU logo and the text 'professionalism in programming'. A search bar is present with the text 'Search in Book Reviews'. Below the search bar, there is a description of the ACCU review process and a search form. The search results show 268 results where the title contains 'java'. The results list several books, including 'A Programmer's Guide to Java Certification', 'Advanced Techniques for Java Developers', 'Beginning Java 2 SDK 1.4 Edition', 'Concurrent Programming in Java 2ed', and 'Core Java 1.1 - Volume 1 Fundamentals'. The page also includes an advertisement for 'Because C++' and a membership section with a 'Join' button. The browser's address bar shows the URL 'https://accu.org/index.php?module=bookreviews&func=search'. The Windows taskbar at the bottom shows various open applications and the system clock.

Facebook x star wars jabba - Google x Europäische Krankenvers x TU Wien - Google Drive x ACCU :: Book Reviews Se x

https://accu.org/index.php?module=bookreviews&func=search

SWAROVSKI Eclipse F ★ Bookmarks http://redirect.kaspers On-лайн ресурси YouTube Cayin C5 Portable HIF Using MATLAB and A Google Translate Клуб на собственици BGAUDIO.ORG Audio Visitenkarte von Tray »

**accu** professionalism in programming **accu 2017**

ACCU Menu

- Conferences
- Journals
- Book Reviews
- Local Groups
- Community
- Membership
- About

Advertisement

Your Ad Here

Over 15000 software professionals visit this site every month. To advertise in this space contact - ads@accu.org

Search in Book Reviews

The ACCU passes on review copies of computer books to its members for them to review. The result is a large, high quality collection of book reviews by programmers, for programmers. Currently there are 1913 reviews in the database and more every month. Search is a simple string search in either book title or book author. The full text search is a search of the text of the review.

Search for   [View all alphabetically](#)

By title ☒

By author ☐

Full Text ☐

268 results where title contains 'java':

< << 1 2 3 4 5 6 7 8 9 10 >> >|

- [A Programmer's Guide to Java Certification](#) by Khalid Mughal&Rolf Rasmussen - **Highly recommended**
- [Advanced Techniques for Java Developers](#) by D. Berg&J. Fritzinger - **Highly recommended**
- [Beginning Java 2 SDK 1.4 Edition](#) by Ivor Horton - **Highly recommended**
- [Concurrent Programming in Java 2ed](#) by Doug Lea - **Highly recommended**
- [Core Java 1.1 - Volume 1 Fundamentals](#) by C Horstmann&G Cornell - **Highly recommended**

Advertisement

**Because C++**

Get an IDE

Membership

**Join**

Login

Username:

Password:

☐ Remember me

Screenshot\_2016-....png ^ CsellXEWYAAcKBm.jpg ^ 87a3ba6da0c2520f....jpg ^ 1755caa8e3bf5e42....jpg ^ slave\_leia\_and\_ja....jpeg ^ Olivia\_and\_jabba\_b....jpg ^ Show all x

Ask me anything

ACCU ... Skype Kasper... Movie... Java is...

ENG INTL 00:03 19-Nov-16

3. Working with **2D** arrays in order to display,  
organise and calculate **our common world** better

## 4. Career chances and project work: the **key** for *success*

1. Engineers without Borders Austria:  
<http://www.iog-austria.at/welcome/?L=1> for now they are making garden table with solar roof together with refugees, **they will need your help** for making computer program to record the data from the sensors in it
2. Infonova: <https://www.infonova.com/en/index.html> Ecosystems for Smart Life (They are working with Engineers without Borders as well, especially **Philipp Pichler**, I spoke with him personally **so we can** communicate with him as well in the future:  
[https://www.xing.com/profile/Philipp\\_Pichler10](https://www.xing.com/profile/Philipp_Pichler10) his email is [philipp.pichler@infonova.com](mailto:philipp.pichler@infonova.com)
3. <https://www.fabasoft.com/en/> Fabasoft: Cloud Solutions for big companies and more
4. [http://www.ranorex.com/?gclid=CMGPuLW\\_s9ACFXAo0wod0mEOxQ](http://www.ranorex.com/?gclid=CMGPuLW_s9ACFXAo0wod0mEOxQ)  
Ranorex: Automatic Testing of your code (but only when your projects become really BIG, not for now)
5. <http://www.karriere.at/jobs/4646173> Allianz needs young specialist in Java. English and German in B2 Level and some IT Degree are a must!

# 5.The Homework!-> the game 4 in a row (Old but Gold)

```
import java.util.Scanner;
public class Main{
//global variables
//game board
//creates scanner
Public static void main(String[] args){
//creates board
//tells player how to play
//displays board
//creates boolean to determine status of game
//main game loop
//activates player 1s turn, then prints board
//determines if player 1 has won
//sets flag to false so loop is not repeated if player 1 won
//break to skip player 2s turn if won
//activates player 2s turn, then prints board
//determines if player 1 has won
//sets flag to false so loop is not repeated if player 2 won
// break for consistency
public static void CreateBoard() {
//fills board with '.' for the width and height
public static void PrintBoard() {
//prints the board
public static void DropX(){
//creates a counter
//shows whos turn
//gets input
//checks to see if space is blank, puts X there if it is
//breaks loop after placing
//if space isn't blank, checks to see if one above is
//puts X if blank
//breaks loop after placing
//adds one to counter if the space wasn't blank, then loops again
//checks to see if at end of column
public static void DropO(){
```



```
//creates a counter
//shows whos turn
//gets input
//checks to see if space is blank, puts O there if it is
//breaks loop after placing
//if space isn't blank, checks to see if one above is
//puts O if blank
//breaks loop after placing
//adds one to counter if the space wasn't blank, then loops again
//checks to see if at end of column
public static boolean CheckXHorizontal(){
//creates boolean to act as flag
//creates counter
//goes through board horizontally
//if it finds an X, add 1 to counter
// if next piece is not an X, set counter to 0
//if counter is greater or equal to 4, player wins
public static boolean CheckXVertical(){
//creates boolean to act as flag
//creates counter
//goes through board vertically
//if it finds an X, add 1 to counter
// if next piece is not an X, set counter to 0
//if counter is greater or equal to 4, player wins
public static boolean CheckOHorizontal(){
//creates boolean to act as flag
//creates counter
//goes through board horizontally
//if it finds an O, add 1 to counter
// if next piece is not an O, set counter to 0
//if counter is greater or equal to 4, player wins
public static boolean CheckOVertical(){
//creates boolean to act as flag
//creates counter
//goes through board vertically
//if it finds an O, add 1 to counter
// if next piece is not an O, set counter to 0
//if counter is greater or equal to 4, player wins
```

```
public static boolean CheckXDiagonalForward(){
//flag
//counter
//check boolean
//checkers
//goes through until an X is found
//if X is found, add one to counter and go into loop
//goes through diagonally looking for Xs
//if X is found, add 1 to counter
//adds 1 to checkers
//if outside of board, break
//if counter is greater or equal to 4, player wins
//resets counter and checkers
public static boolean CheckODiagonalForward(){
//flag
//counter
//check boolean
//checkers
//goes through until an O is found
//if O is found, add one to counter and go into loop
//goes through diagonally looking for Os
//if O is found, add 1 to counter
//adds 1 to checkers
//if outside of board, break
//if counter is greater or equal to 4, player wins
//resets counter and checkers
public static boolean CheckXDiagonalBack(){
//flag
//counter
//check boolean
//checkers
//goes through until an X is found
//if X is found, add one to counter and go into loop
//goes through diagonally looking for Xs
//if X is found, add 1 to counter
//adds 1 to checkers
//if outside of board, break
//if counter is greater or equal to 4, player wins
//resets counter and checkers
public static boolean CheckODiagonalBack(){
//flag
//counter
//check boolean
//checkers
//goes through until an O is found
//if O is found, add one to counter and go into loop
//goes through diagonally looking for Os
//if O is found, add 1 to counter
//adds 1 to checkers
//if outside of board, break
//if counter is greater or equal to 4, player wins
//resets counter and checkers
```

# As a dessert I give you the whole final of the program

```
public static boolean CheckX(){
    //creates flag
    boolean flag = true;
    //checks all Xs at once, for clearer main loop
    if(!CheckXVertical() || !CheckXHorizontal() || !CheckXDiagonalBack() || !CheckXDiagonalForward()){
        flag = false;
    } return flag;
}

public static boolean CheckO(){
    //creates flag
    boolean flag = true;
    //checks all Os at once, for clearer main loop
    if(!CheckOVertical() || !CheckOHorizontal() || !CheckODiagonalBack() ||
!CheckODiagonalForward()){
        flag = false;
    }
    return flag;
}
}
```

Java advices: Don't  
**forget** to make  
your homework!

