Workshop Software Development

Tim Beurskens

Honors Academy – Empowerment for Health & Wellbeing
Eindhoven University of Technology

Contents - 1

Contents - 2

Workflow

Integrated Development Environment (IDE)

- Jetbrains IntelliJ IDEA (Java)
- Microsoft Visual Studio (C#) + Jetbrains ReSharper
  - (.net Desktop Development)

Coding Style

- (Usually) handled by IDE
- Makes code readable for yourself and other developers
- Most languages have a (well documented) coding standard

Version Control System

- Keeps track of changes in files
- Allows multiple developers to work on a single file
- In our case GIT, why?
  - Widely used
  - Free hosting on https://www.github.com/ as a student
  - “easy” to learn
  - Visual Studio has a Github extension

Exercise

1. Install Microsoft Visual Studio 2017 Community (+GIT)
2. Clone the GIT repo from https://github.com/timbeurskens/KinectTutorial
3. Check out in the branch named “exercise-1-{your_name}”
4. Create a program that computes the GCD of two integers
   - Input from stdin and output to stdout (Console)
5. Commit your changes and push them to the remote branch

Exercise - Hint

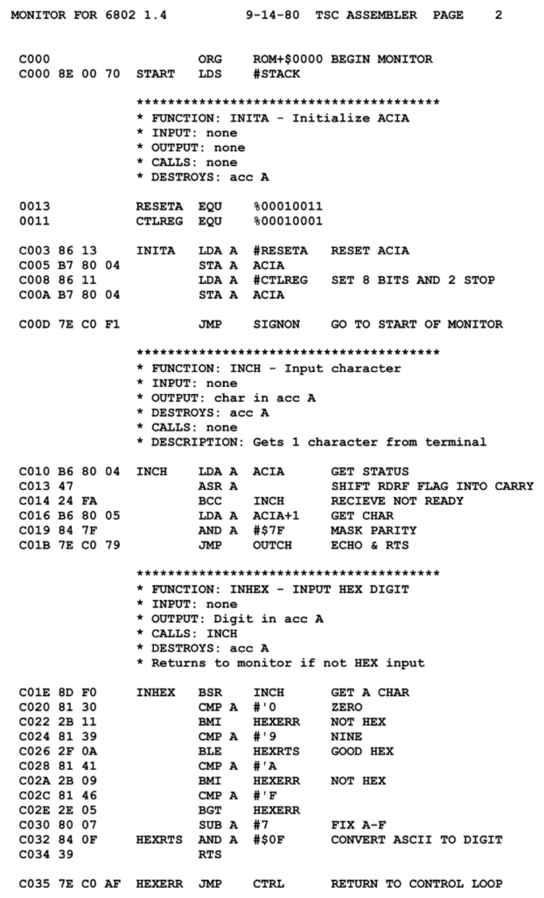
$$
\begin{array}{lcllcl}
gcd(a,b) & = & \text{if} & a=0 & \rightarrow & b \\
 & & [] & a>0 & \rightarrow & gcd(b \bmod a, a) \\
 & & \text{fi} & & &
\end{array}
$$

Programming paradigms

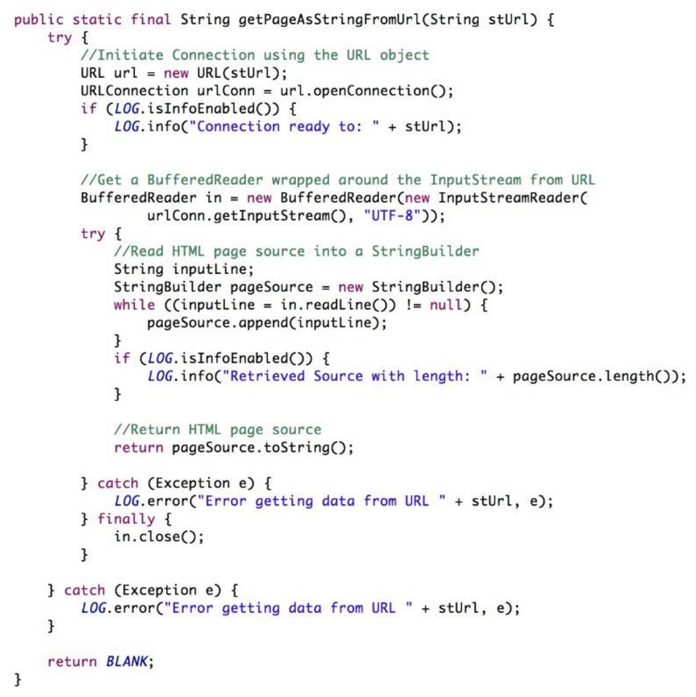
Why do we have so many programming languages

- Most languages have the same expressive power (Church-Turing Thesis)

Assembly vs Java



```
MONITOR FOR 6802 1.4          9-14-80  TSC ASSEMBLER  PAGE    2

 C000                  ORG    ROM+$0000 BEGIN MONITOR
 C000 8E 00 70  START  LDS    #STACK

                ****************************************
                * FUNCTION: INITA - Initialize ACIA
                * INPUT: none
                * OUTPUT: none
                * CALLS: none
                * DESTROYS: acc A

 0013           RESETA EQU    %00010011
 0011           CTLREG EQU    %00010001

 C003 86 13     INITA  LDA A  #RESETA   RESET ACIA
 C005 B7 80 04         STA A  ACIA
 C008 86 11            LDA A  #CTLREG   SET 8 BITS AND 2 STOP
 C00A B7 80 04         STA A  ACIA

 C00D 7E C0 F1         JMP    SIGNON    GO TO START OF MONITOR

                ****************************************
                * FUNCTION: INCH - Input character
                * INPUT: none
                * OUTPUT: char in acc A
                * DESTROYS: acc A
                * CALLS: none
                * DESCRIPTION: Gets 1 character from terminal

 C010 B6 80 04  INCH   LDA A  ACIA      GET STATUS
 C013 47               ASR A            SHIFT RDRF FLAG INTO CARRY
 C014 24 FA            BCC    INCH      RECIEVE NOT READY
 C016 B6 80 05         LDA A  ACIA+1    GET CHAR
 C019 84 7F            AND A  #$7F      MASK PARITY
 C01B 7E C0 79         JMP    OUTCH     ECHO & RTS

                ****************************************
                * FUNCTION: INHEX - INPUT HEX DIGIT
                * INPUT: none
                * OUTPUT: Digit in acc A
                * CALLS: INCH
                * DESTROYS: acc A
                * Returns to monitor if not HEX input

 C01E 8D F0     INHEX  BSR    INCH      GET A CHAR
 C020 81 30            CMP A  #'0       ZERO
 C022 2B 11            BMI    HEXERR    NOT HEX
 C024 81 39            CMP A  #'9       NINE
 C026 2F 0A            BLE    HEXRTS    GOOD HEX
 C028 81 41            CMP A  #'A
 C02A 2B 09            BMI    HEXERR    NOT HEX
 C02C 81 46            CMP A  #'F
 C02E 2E 05            BGT    HEXERR
 C030 80 07            SUB A  #7        FIX A-F
 C032 84 0F     HEXRTS AND A  #$0F      CONVERT ASCII TO DIGIT
 C034 39               RTS

 C035 7E C0 AF  HEXERR JMP    CTRL      RETURN TO CONTROL LOOP
```



```java
public static final String getPageAsStringFromUrl(String stUrl) {
    try {
        //Initiate Connection using the URL object
        URL url = new URL(stUrl);
        URLConnection urlConn = url.openConnection();
        if (LOG.isInfoEnabled()) {
            LOG.info("Connection ready to: " + stUrl);
        }

        //Get a BufferedReader wrapped around the InputStream from URL
        BufferedReader in = new BufferedReader(new InputStreamReader(
                urlConn.getInputStream(), "UTF-8"));
        try {
            //Read HTML page source into a StringBuilder
            String inputLine;
            StringBuilder pageSource = new StringBuilder();
            while ((inputLine = in.readLine()) != null) {
                pageSource.append(inputLine);
            }
            if (LOG.isInfoEnabled()) {
                LOG.info("Retrieved Source with length: " + pageSource.length());
            }

            //Return HTML page source
            return pageSource.toString();

        } catch (Exception e) {
            LOG.error("Error getting data from URL " + stUrl, e);
        } finally {
            in.close();
        }

    } catch (Exception e) {
        LOG.error("Error getting data from URL " + stUrl, e);
    }

    return BLANK;
}
```

Brainf*ck

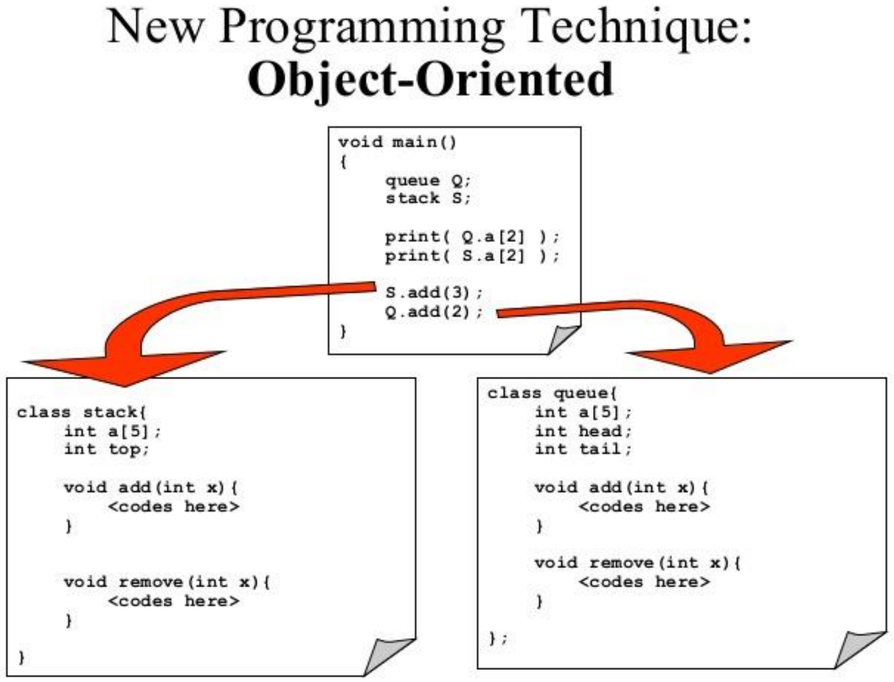
```
++++++++++[>+++
++++>+++++++++
+>+++>+<<<<-]>+
+.>+.+++++++..+
++.>++.<<++++++
+++++++++.>.++
+.------.------
--.>+.>.
```

Object Oriented Programming

From Microsoft's MSDM website:

"The terms *class* and *object* are sometimes used interchangeably, but in fact, classes describe the *type* of objects, while objects are usable *instances* of classes. So, the act of creating an object is called *instantiation*. Using the blueprint analogy, a class is a blueprint, and an object is a building made from that blueprint."

Object-Oriented Programming

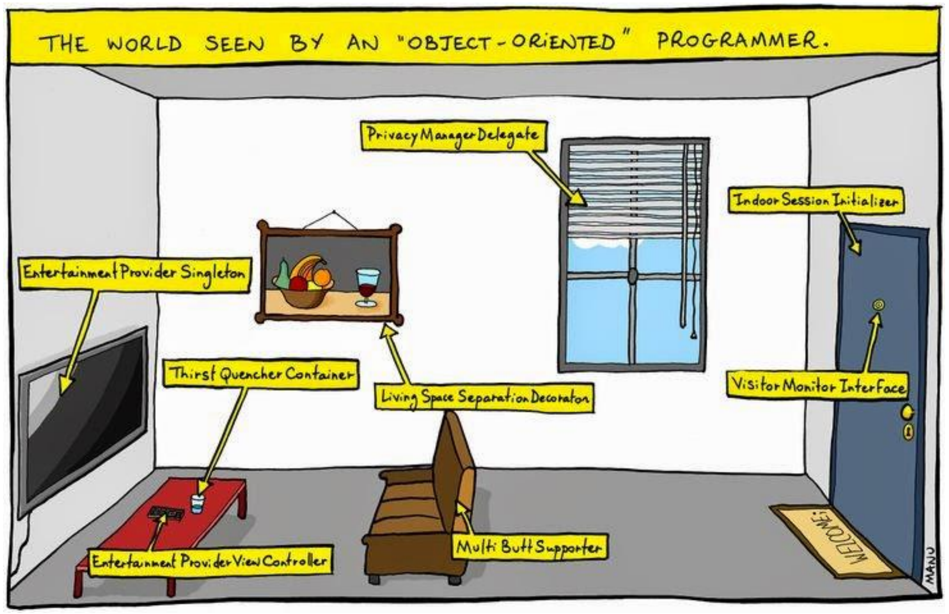


New Programming Technique:
Object-Oriented

```
void main()
{
    queue Q;
    stack S;

    print( Q.a[2] );
    print( S.a[2] );

    S.add(3);
    Q.add(2);
}
```

```
class stack{
    int a[5];
    int top;

    void add(int x){
        <codes here>
    }


    void remove(int x){
        <codes here>
    }
}
```

```
class queue{
    int a[5];
    int head;
    int tail;

    void add(int x){
        <codes here>
    }

    void remove(int x){
        <codes here>
    }
};
```


THE WORLD SEEN BY AN "OBJECT-ORIENTED" PROGRAMMER.
Privacy Manager Delegate
Indoor Session Initializer
Entertainment Provider Singleton
Thirst Quencher Container
Living Space Separation Decorator
Visitor Monitor Interface
Entertainment Provider View Controller
Multi Butt Supporter
WELCOME
MANU

Exercise

- Try to represent a 3D vector as an object
- Create a method that computes the angle between 2 vectors
- Commit your changes to the branch:
  - “exercise-2-{your_name}”

$$\cos(\theta) = \frac{\vec{x} \cdot \vec{y}}{||\vec{x}||\ ||\vec{y}||}$$

Event-Driven Programming

Multithreading

- + Multiple processes running at the same time
- + Large computations are not blocking the main thread
- - Very difficult to synchronize between threads
- - Difficult to debug

Test-Driven Development

1. Build a general structure of your program:
   - Classes
   - Public / shared methods
   - Public / shared variables
2. Create documentation for all public methods, variables, classes
3. Write automated test cases for every possible case you can think of
4. Write code for the methods created in step 1
5. Test your code using the test cases created in step 3

Microsoft Kinect

Connecting the sensor

Listening for data

Data Processing

:(

Your PC ran into a problem and needs to restart. We're just collecting some error info, and then we'll restart for you.

If you'd like to know more, you can search online later for this error: ALWAYS_LOOK_ON_THE_BRIGHT_SIDE_OF_LIFE

References & Further reading

- Git for Windows: https://git-scm.com/download/win
- Learn git: https://try.github.io/
- Kinect tutorials: http://kinect.github.io/tutorial/index.html
- Jetbrains ReSharper: https://www.jetbrains.com/resharper/
- Jetbrains IntelliJ IDEA (Java): https://www.jetbrains.com/idea/
- Microsoft Visual Studio: https://www.visualstudio.com
- Microsoft Kinect SDK: https://developer.microsoft.com/en-us/windows/kinect/tools
- Object Oriented Programming concepts: https://docs.oracle.com/javase/tutorial/java/concepts/
- Test Driven Development: http://agiledata.org/essays/tdd.html
- C# Tutorial: https://www.tutorialspoint.com/csharp/
- Java code conventions: http://www.oracle.com/technetwork/java/codeconventions-150003.pdf
- C# code conventions: https://msdn.microsoft.com/en-us/library/ff926074.aspx
- C# code conventions (unofficial): http://www.dofactory.com/reference/csharp-coding-standards