

PSEUDOCODE AND FLASH PLATFORM

Model and Create an Advanced Image Editor

AGENDA

- What is pseudocode
- Best practices
- Agile modeling
- Pseudocode syntax
- From requirements to pseudocode (and source code!)
- A real life example, an image editor

WHAT IS PSEUDOCODE

WHAT IS PSEUDOCODE

- Is a compact and informal description of an algorithm or of an entire program
- Is a way to improve communication between team members
- Is a way to improve programming skills of junior developers
- Is a very informal way to design a software or part of it
- Is a placeholder for discussions and clarifications
- It's a compact way to describe the business logic of an application

WHAT IS PSEUDOCODE

- The lack of formalism is one of the strength of pseudocode
- The main purpose is to make code (and software development in general) understandable
- Too often the code looks like “brain fuck” (<http://en.wikipedia.org/wiki/Brainfuck>)

```
+++++ +++++
[
to 70/100/30/10
  > +++++ ++
  > +++++ +++++
  > +++
  > +
  <<<< -
]
```

```
initialize counter (cell #0) to 10
use loop to set the next four cells

  add 7 to cell #1
  add 10 to cell #2
  add 3 to cell #3
  add 1 to cell #4
decrement counter (cell #0)
```

WHAT IS PSEUDOCODE

- Pseudocode usually omits details that are not essential for human understanding
- The intent is to make easier exchanging ideas about the key principles of an algorithm
- As a result also the code should be much more clear because the usage of pseudocode brings to a “design first” approach
- Designing first the code brings to excellent results and less legacy code

WHAT IS PSEUDOCODE

- It's NOT an easiest version of UML
- It's NOT a language with specific rules to learn
- It's NOT a time waster if used in the proper way, when writing pseudocode
 - Don't be too much detailed
 - Don't be too much related to a specific language
- It's a good starting point to document your code (if needed!)

WHAT IS PSEUDOCODE

- A snippet of pseudocode should look like this

```
INIT the hours worked by an employee
READ maximum allowed hours FROM web service
IF workedHours > maxAllowed THEN
    SHOW overtime message
ELSE
    SHOW regular time message
ENDIF
```

- The reason why there are some words written in capital letters is to improve readability
- It's easy to create a function or a method from this snippet without mistakes or even better without any misunderstanding

BEST PRACTICES

BEST PRACTICES

- Pseudocode strikes a precarious balance between the understandability and informality of English and the precision of code
- The first temptation of every programmer is to write an algorithm in code but it means spend time defining a lot of details
- The goal of writing pseudocode is to provide a high-level description of an algorithm which facilitates analysis, eventual coding and the production of the documentation

BEST PRACTICES

- When writing pseudocode a top down approach help to identify clearly each part of a program
- The usage pseudocode doesn't imply to model with it each part of a program
- Pseudocode helps to understand on which part of a program you need to make a deeper analysis and to make more precise estimates
- Pseudocode is a way to understand which part of a program are still not well defined

BEST PRACTICES

- Keep the pseudocode at an “high-level” make it easier to read and more useful
- The boundaries outlined by the words “high-level” depend on the audience
- Always avoid to belabor the obvious, it’s not critical to specify the data type of a variable or to set up the counter to use in a loop when writing pseudocode.

```
SET the current user
FOR the list of employees
  IF match found THEN
```

```
    CALL reduce the hours on the found user
```

```
  ENDIF
```

```
ENDFOR
```

BEST PRACTICES

- Use English words like IF, FOR, THEN, SET, etc. that are also standards in most of the programming languages to be understandable by technical and not technical guys
- Consider the context in which you are and avoid to specify twice the purpose of an algorithm

```
PROCEDURE reduceHours RETURNING void
    reduce the amount of hours worked
    SET the current user
    FOR the list of employees
        IF match found THEN
            CALL reduce the hours on the found user
        ENDIF
    ENDFOR
ENDPROCEDURE
```

BEST PRACTICES

- The previous snippet can be shorter and look like this

```
PROCEDURE reduceHours RETURNING void
  SET the current user
  FOR the list of employees
    IF match found THEN

        // reduce the hours on the found user

    ENDIF

  ENDFOR
ENDPROCEDURE
```

BEST PRACTICES

- Keep always in mind that is not needed to define through pseudocode an entire program
- Don't pretend to define every facet of a program before starting to code, pseudocode is informal and very practical so avoid to make it an additional cost
- Use pseudocode to start the process of translating software requirements into a shape that can be understandable also for domain experts
- It's very hard to keep the pseudocode easy to read because actually it's text, use indentation and capital letters to focus the attention on the right part of it

AGILE MODELING

AGILE MODELING

- The lack of formalism and it's practical approach make pseudocode very close to the agile modeling practices
- The key principles of agile can be summarized as following

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

- Pseudocode supports the idea of iterative refinement so it's easy to keep it connected with the requirements that usually change a lot during the development

AGILE MODELING

- The steps that brings from pseudocode to source code represent the iterative approach suggested from the agile modeling
 1. Top level pseudocode definition
 2. Breakdown of the top level pseudocdoe
 3. Detailed (not too much!) pseudocode
 4. Pseudocode refinement
 5. Source code
- The points 2, 3 and 4 represent the iterative approach, the end result (i.e. the source code) represents the end of the refinement and even more it's actually a piece of working software!

AGILE MODELING

- The successive refinement in small steps allows to check the software design driving it to lower levels of detail
- One of the benefit is to catch high-level errors at the highest level, mid-level errors at the middle level, and low-level errors at the lowest level before any of them becomes a problem
- This approach let developers starting code sooner because the principle it's not to define all the components of a software but "just enough" to move on and work on the first release of a potentially shippable version of the software

AGILE MODELING

- It's quite easy to evolve a method from a real basic version to a much more detailed one

```
PROCEDURE login user against LDAP
    CALL the web service for authentication USING username, password
ENDPROCEDURE
```

```
PROCEDURE login user against LDAP
    ARGUMENTS username , password

    IF username and password are not empty THEN
        CALL the web service for authentication USING username, password
    ELSE
        SHOW an error message
    ENDIF
ENDPROCEDURE
```

- Both versions work fine but the second one is almost ready to be source code

AGILE MODELING

- Performing an iterative refinement of the pseudocode is one way to hit the right audience at the right moment
- Too much details can be confusing (or even better annoying) for architects or domain expert but can help a lot developers
- The iterative refinement of the pseudocode helps to make short term plans and to identify issues or delays in time
- The text nature of pseudocode makes it really open to changes

PSEUDOCODE SYNTAX

PSEUDOCODE SYNTAX

- As the name itself suggests the pseudocode generally does not follow any specific syntax rule and details
- The overall concept is to describe an algorithm in a form that a developer can easily translate into code and analysts can quickly check
- One of the most complete syntax definition has been made by Stuart Garner, he created a library for *NoteTab* in order to write down pseudocode

PSEUDOCODE SYNTAX

- The usage of pseudocode brings to the definition of some common terms because of its “structured” nature
- The structured part of pseudocode is a notation for representing some specific structured programming constructs like
 - CLASS
 - INTERFACE
 - PROCEDURE
 - IF-THEN-ELSE
 - FOR
 - SWITCH CASE
- New constructs can be added as needed, the important thing is to keep it clear, logically grouped and short as much as possible

FROM USER STORIES TO PSEUDOCODE

USER STORY → PSEUDOCODE

- Most of the time software requirements are expressed as user stories
- A user story is made up by one or more sentences in the everyday or business language of the end user that captures what the user wants to achieve
- User stories could also be written by developers to express non-functional requirements

USER STORY → PSEUDOCODE

- Let's imagine to working on a login form of an application that has to support sign-in through the usage of *openID* and *connect* (i.e. facebook)
- The requirement is fully defined by the following user story

Login

As a user I can login into the application using an open id or connect standards (Facebook, Twitter, Google) so that I can avoid the registration process

USER STORY → PSEUDOCODE

- It can be translated into the following snippet of pseudocode

```
PROCEDURE login without registration
  ARGUMENTS username , password , external service

  SET store the username and password
  CASE external service IS
    facebook THEN :
      CALL facebook API wrapper

    google THEN :
      CALL google API wrapper

    OTHERS THEN :
      CALL authenticate through the web service

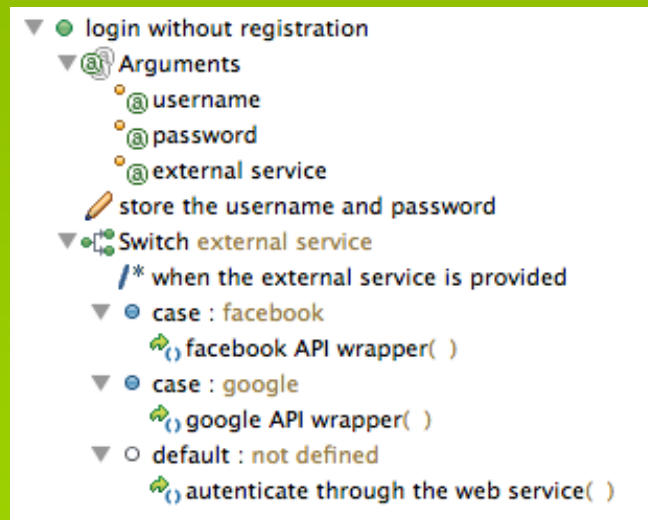
  ENDCASE

ENDPROCEDURE
```

- It is still readable and tightly coupled with the requirement

USER STORY → PSEUDOCODE

- Most of the people that look to pseudocode for the first time are quite disappointed because the first impression is to write twice
- The previous snippet has been generated starting from the following diagram



- The input needed is very short so this activity doesn't start to be time waster

USER STORY → PSEUDOCODE

- From the same diagram it's possible to generate the source code for the following programming languages
 - ActionScript
 - Java
 - PHP
 - JavaScript
 - ... more coming

USER STORY → PSEUDOCODE

- ActionScript 3.0 sample output

```
package {  
  
    /**  
     * just a placeholder for the presentation  
     */  
    public class DummyClass{  
  
        /**  
         * NOTE: login without registration  
         */  
        public function RENAME_ME(username:/*type*/, password:/*type*/,  
                                   external service:/*type*/):void {  
  
            // SET store the username and password  
            switch (external service) {  
  
                case facebook:  
                    facebook API wrapper();  
                    break;  
  
                case google:  
                    google API wrapper();  
                    break;  
  
                default:  
                    authenticate through the web service();  
                    break;  
  
            }  
        }  
    }  
}
```

USER STORY → PSEUDOCODE

- Java sample output

```
/**
 * just a placeholder for the presentation
 */
public class DummyClass{

    /**
     * NOTE: login without registration
     */
    public void RENAME_ME(/*type*/ username, /*type*/ password,
                          /*type*/ external service) {

        // SET store the username and password
        switch (external service) {

            case facebook:
                facebook API wrapper();
                break;

            case google:
                google API wrapper();
                break;

            default:
                authenticate through the web service();
                break;

        }

    }

}
```

USER STORY → PSEUDOCODE

- PHP sample output

```
<?php
namespace ;

public class DummyClass{

    public function __construct(){}
    public function __destruct(){}

    // NOTE: login without registration
    public function RENAME_ME( $username, $password, $external service) {

        // SET store the username and password
        switch (external service) {

            case facebook:
                facebook API wrapper();
                break;

            case google:
                google API wrapper();
                break;

            default:
                authenticate through the web service();
                break;

        }

    }

}
?>
```

USER STORY → PSEUDOCODE

- JavaScript sample output

```
DummyClass = function() {  
  
    /**  
     * NOTE: login without registration  
     */  
    this.RENAME_ME = function(username, password, external service) {  
  
        // SET store the username and password  
        switch (external service) {  
  
            case facebook:  
                facebook API wrapper();  
                break;  
  
            case google:  
                google API wrapper();  
                break;  
  
            default:  
                authenticate through the web service();  
                break;  
  
        }  
  
    }  
  
};
```

USER STORY → PSEUDOCODE

Let's try by ourselves, just open eclipse and install the software that generated the pseudocode from the following URL

<http://apdt.gnstudio.biz/updatesite/beta/>

Public release planned
1st week of July 2011

RESOURCES

USEFUL LINKS

- <http://www.unf.edu/~broggio/cop2221/2221pseu.htm>
- <http://en.wikipedia.org/wiki/Pseudocode>
- <http://www.minich.com/education/wyo/stylesheets/pseudocode.htm>
- <http://userpages.wittenberg.edu/bshelburne/.../Algorithms.htm>
- <http://www.seattlecentral.edu/~ymoh/mic110vb/pseudocode.htm>
- <http://www.agilemodeling.com/>
- <http://riarockstars.com/2011/04/04/user-stories-pseudocode-and-agile-development/>

BOOKS AND LECTURES

- **The Art of LEGO® MINDSTORMS® NXT-G Programming**
Chapter 7. the WallFollower program: navigating a maze
- **PROGRAMMING IN COBOL/400: 2nd Edition**
Chapter 5. DESIGNING STRUCTURED PROGRAMS
- **Code Complete, Second Edition**
Chapter 9. The Pseudocode Programming Process

THANKS FOR ATTENDING!
FOLLOW US: @ENVISION_APDT
FREE LICENSE CODE: FATCKUCF