

3815ICT-Software Engineering Workshop 2

Activity 1

Visit the Web page google.github.io/styleguide/cppguide.html. Explore the page and review it with a critical mindset on why would someone publish this content.

Consider writing the code of a class Person with attributes (properties), name, first_name, date_of_birth and age. Investigate how would you code such a class in the following Object-Oriented Languages: Java, C++, Objective-C, and python. Investigate how would you create an array of objects of the class Person, and how would you create objects that fill the array as well as how to print the elements of the array with the average age at the end.

We will provide you with the bare-bones of a C++ class Person. Criticize that implementation.

Discuss meaningfully how these different programming languages support principles of information hiding (reducing coupling) and enabling encapsulation.

Response 1

There is redundancy in the current state of class Person. It is possible to extrapolate a person's age from their date of birth so the data is essentially being stored twice within each object. Additionally, having one 'name' and one 'first_name' member is quite confusing. They should be 'first_name' and 'last_name' or something clearer. Finally, my preference in variable name writing is not to use an underscore to separate words. 'firstName' would be my preferred method of expressing variable names.

Besides this, the languages Java, Python, C++ and Objective-C do have various similarities and differences in their approaches to coupling and encapsulation.

One glaring difference visible with Python however is in its information hiding standards. There is no way to declare a variable as private in Python Declaration of a private variable is one aspect of low couple which ensures that information is segregated in an effective manner. Python uses a simple 'convention' in which any member declared in the class with an underscore preceding it is to be considered private from any other programmer's perspective. However, there is nothing in place within the language itself to actually enforce the privacy.

There are various ways in Objective-C to declare a member private and Java/C++ have a very blatant declaration for public and private variables. The default member in Java is set to private and any public declaration will remain thereafter. C++ allows for declaration of each individual member/method. Generally the only way to access members within the classes, a specific method is created to facilitate the data exchange. This privacy should prevent other classes to call members directly and ideally, in order to prevent excessive coupling, classes would not be calling on information within another function. This leads into the concept of encapsulation in which each class functions appropriately within its own sphere.

Activity 2

Investigate the use of doxygen (www.stack.nl/~dimitri/doxygen/). Aim to install it and produce automatically the documentation of the class Person for one of the programming languages of the previous activity. Configuring a Doxyfile for your project (folder with source code files) may be laborious with a plain editor. You can use the doxywizard which is a helpful GUI to achieve this. You can see a video of using this interface to doxygen and even running doxygen from the GUI at www.youtube.com/watch?v=mgVgZjaeNkw. Write 2 benefits of using a system for the automatic generation of documentation.

Response 2

The benefits of automatic documentation generation essentially speak for themselves. There are surely drawbacks in that perhaps manual documentation creation though laborious will provide an opportunity to check over one's current work and perhaps find mistakes or shortcuts in implementation. In terms of document generation, it can be laborious and dry so one major benefit is to automate that process in order to focus more on implementation itself.

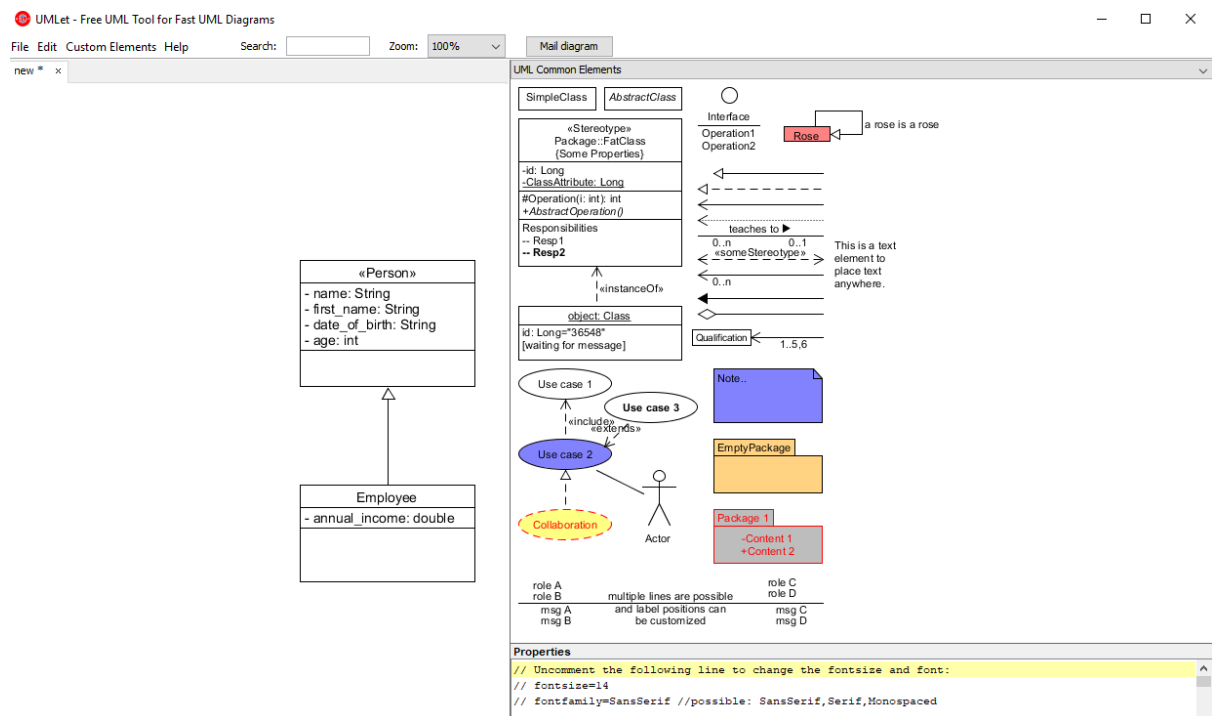
Another benefit is clearly the time saved, assuming an implementation is created, tested and bug free, automating the process of documentation will save a developer valuable time. Again, the risks involved would be mistakes in the original implementation being carried over to the documentation, however, diligence while programming will ensure this is less likely to happen.

Activity 3

Download the shareware UMLet (www.umlet.com). Use it to draw a class diagram of the class Person in the previous activities. Then, assume that there is a class Employee. All employees are people with the additional property that they have annual income. Represent the class Employee and the inheritance relationship to Person using UMLet. Investigate what other alternative tools would enable you to draw an object model (a class diagram) using the basic elements of the Unified Modelling Language (UML).

Response 3

Even the most basic tools such as Powerpoint would allow a sufficient UML to be drafted. There are more dedicated tools out there such as UMLet but the concept of drafting UMLs is not a difficult one to express with basic shapes and lines. One excellent alternative to UMLet is Microsoft Visio which is one of these more dedicated pieces of software with a range of different diagram templates being catered to. Another example of a dedicated charting software is Lucidchart. The advantage of UMLet is that it is free and open-source.



Activity 4

Make a summary of the main points regarding

1. Event-driven programming (look at the page “Introduction to GUI Building” netbeans.org/kb/docs/java/gui-functionality.html and the section “How Event Handling Works”).
2. GUI design (components and containers) as well as widget behaviour
3. Assistance classes (look at the Tester class for an example)
4. Code factorisation (are all buttons handled the same way? what code is common?)

Then, fix the program so that once a player wins, all remaining (unused) positions of the game become unresponsive (currently they still accept a move).

Response 4

1. Event handling is essentially the process of adding functionality to GUI elements. An event may be when a button is pressed, this involves the click of a mouse on the correct component. The mouse being clicked triggers then event, this has been pre-programmed within the script and the button responds to the trigger by providing the functionality programmed into it. Functionality must be added to each button individually and there are a multitude of different types of inputs which might trigger an event.
2. The design of GUI using this particular IDE is essentially a drag and drop affair where components which are readily available and manipulating the programming behind it provides correct functionality. The widgets such as buttons and panels provide a graphical representation of the objects. In order to use a component it must first be stored inside a container and these are utilized across the Java framework.
3. The assistance class tester contains a small amount of functionality which the main program refers to in order to process a specific scenario. In this case – a win. The class is used solely to support the functionality of the main application and is being stored separately for a number of reasons including reusability, readability and compiling among others.
4. Much like the mathematical concept in which common elements are extracted from an equation in order to minimize the expression, code factorising is the act of extracting commonalities within separate functionalities in order to prevent excessive repeating of code. This means that classes can be reformatted or functions can be created to service multiple elements of functionality.
5. I was able to accomplish the requirement of making the buttons unresponsive within the tic-tac-toe game simply. I created a variable called *winner* and declared it as false. When the **tester** class found there to be a winner, the *winner* variable was set to true.

```
if (Tester.aWin(board).equals("CROSS")) {  
    turnjLabel.setText("CROSS WINS");  
    winner = true;  
}  
  
if (Tester.aWin(board).equals("NOTS")) {  
    turnjLabel.setText("NOTS WINS");  
    winner = true;  
}
```

An additional statement was included within the **judgeButton** function which only runs the function if the *winner* variable is still false.

```
public void judgeButton(javax.swing.JButton aButton, int row, int column) {  
    if (winner != true) {  
        if (aButton.getText().equals(".")) // has not been played  
        {  
            if (turnjLabel.getText().equals("CROSS TURN")) {  
                aButton.setText("X");  
                turnjLabel.setText("\nNOTS TURN\n");  
                turnjLabel.setText("NOTS TURN\n");  
            }  
        }  
    }  
}
```

Activity 5

Provide a reflection on how the points above assist in the elaboration of the Mine Sweeper project. Write two paragraphs, about 7-10 lines each, for what insights you gain (from the activities above) towards the overall project.

Activity 6

The use of Netbeans has opened a large world for me to explore. I am very far behind in this subject at the moment because the only programming I have done is in Python. I have already completed my implementation of a standard minesweeper using Javascript but in order to complete this current workshop I had to go away and study Java, C++, Objective-C and I had to brush up on Python. It has put me behind the 8-ball but I feel much better equipped now to take on coding challenges in various languages and I have a slightly deeper understanding of programming in general because of it. I was very happy to have been able to read and complete a task in Java when I've only just taught myself the basics of 3 new languages over the course of a few days.

Introduction to an IDE which will make game development simpler is nice and I do wish I'd gotten to this workshop sooner but I've been scrambling to focus on the assignment while learning new languages that the workshops have become second priority. It is unfortunate that the information which could have helped me complete the assignment was in the workshop, however, I don't think I would have been able to tackle it without putting those hours into new languages. All in all, I'm only better at programming for it so I'll get it done and I'll potentially rebuild my minesweeper game in Java for the next 2 milestones.