|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Submission Coversheet (All Programmes) | | | |  |  |  |  |
|  |  |  |  |  |  |  |  |
| **Student ID Number**  *(Do not include student name as anonymous marking is implemented)* | Mateusz Utracki-Janeta | | | | | | |
| **Programme Title** | Bsc Computing Technologies | | | | | | |
| **Module Title** | Object Oriented Programming | | | | | | |
| **Module Code** *(listed on Moodle and in LTAFP)* |  | | | | | | |
| **Module Convenor** |  | | | | | | |
| **Coursework Title** | Mechachrome Registration App Critical Comment | | | | | | |
| **Academic Declaration:**  *Students are reminded that the electronic copy of their essay may be checked, at any point during their degree, with Turnitin or other plagiarism detection software for plagiarised material.* | | | | | | | |
| **Word Count** | 1101 | | | **Date Submitted** | | **05/04/2019** | |
|  |  | | |  | |  | |
|  | | | | | | | |

Table of Contents

[1. Use of OOP concepts 3](#_Toc5881397)

[1.1. Object Oriented Programming 3](#_Toc5881398)

[1.2. Classes 3](#_Toc5881399)

[1.3. Inheritance 4](#_Toc5881400)

[1.4. Abstraction 4](#_Toc5881401)

[2. Meeting the design Criteria 4](#_Toc5881402)

[2.1. Enhancements 6](#_Toc5881403)

[2.1.1. Methods and menu class 6](#_Toc5881404)

[2.1.2. Catch statement for edit appointment 6](#_Toc5881405)

[2.1.3. Error testing 7](#_Toc5881406)

[2.1.4. Hash maps 7](#_Toc5881407)

[3. References 7](#_Toc5881408)

# Use of OOP concepts

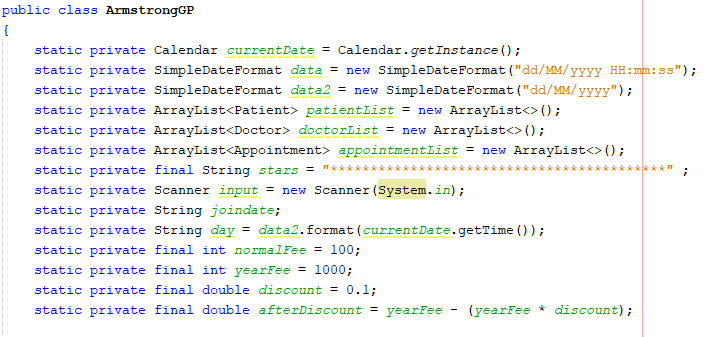
## Object Oriented Programming

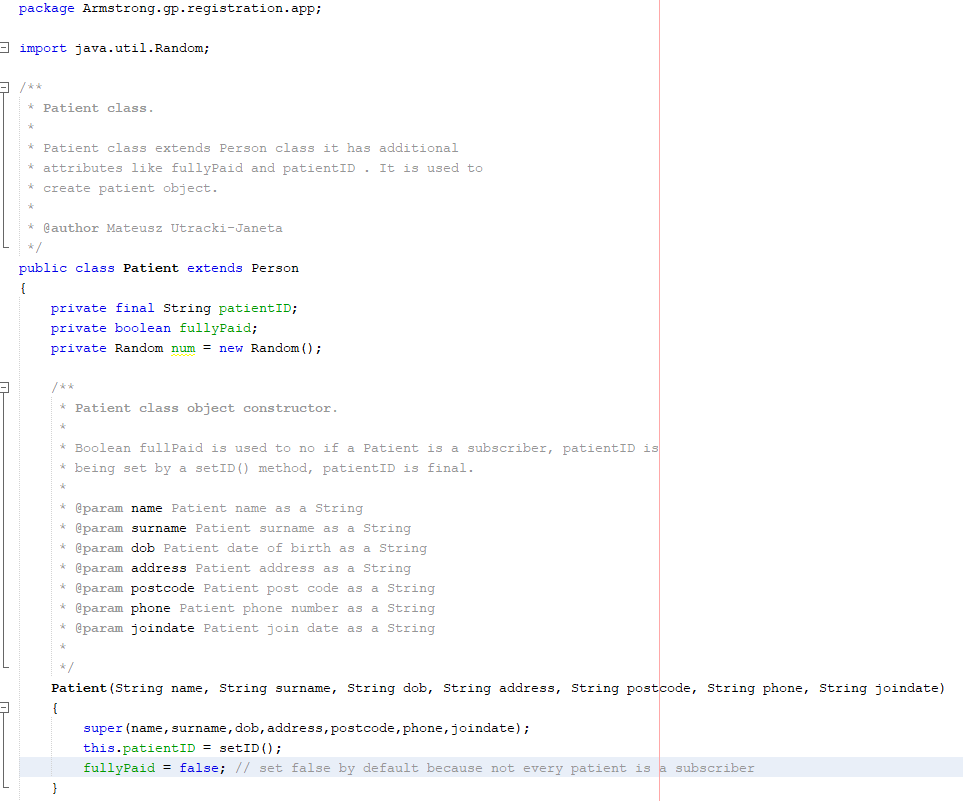
Object oriented programming is a programming paradigm that uses object which has attributes these are often called fields and interact with each other using methods. This concept is very useful because it helps to reduce the complexity and improves the maintainability of the system.

OOP has many interesting characteristics one of the most attractive is the fact that objects can be used to describe real-world entities quite effectively. You can define an employee in a company, like in our case Doctor in the Clinic and a Client who is the Patient. Then we create one object per Doctor or Patient and make them interact using defined behaviours to create an appointment. “That is, the object-oriented approach makes it easier to solve problems, which is the point of writing a program in the first place.” (Lewis 2008 Page 45)

## Classes

Let’s look at the car analogy, car is an object, it has it’s attributes like dimensions, weight, colour and methods like reverse, drive, brake, turn. “A Class is like an object constructor, or a "blueprint" for creating objects.” (W3 Schools) I used classes widely across this project. It is useful to group similar functionalities in one place.





## Inheritance

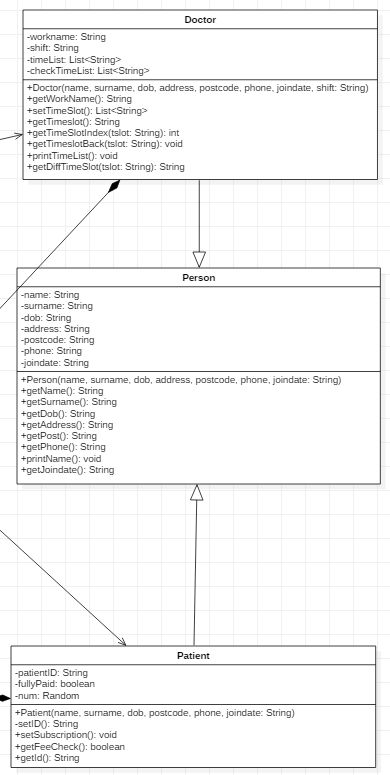
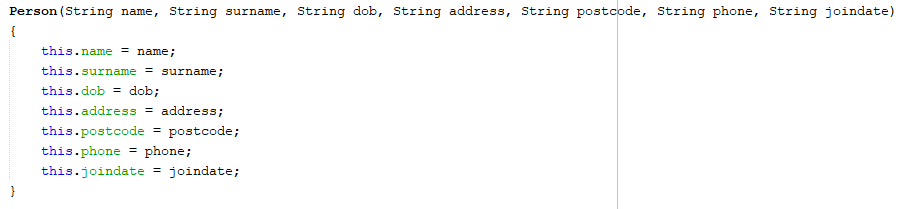
Using the car analogy again we have Coupe, Hatchback, SUV, Sedan type of car classes but we can all agree that they belong to a superclass called Car, in which they share the common attributes and methods, inside themselves they store their unique characteristics. That’s what inheritance is. (Deitel 2015)

Figure 2 Person class constructor

Figure 1

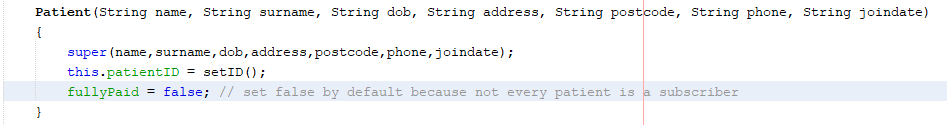


Figure 3 Patient class constructor using Person super constructor

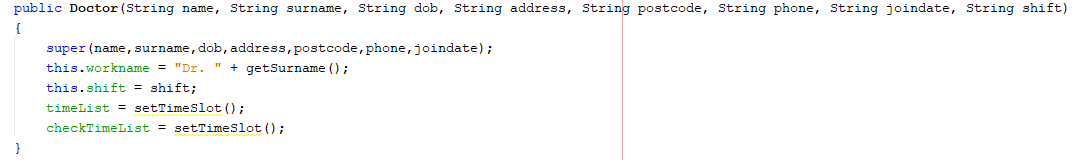
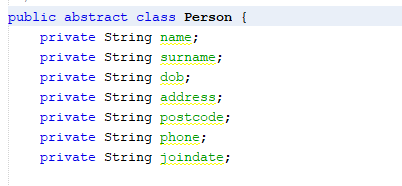


Figure 4 Doctor class constructor using Person super constructor

I used inheritance while creating Doctor and Patient class first of all it saves time you don’t need to rewrite variables and methods again; second thing inheritance can keep your code DRY (Coding practices 2011)

## Abstraction

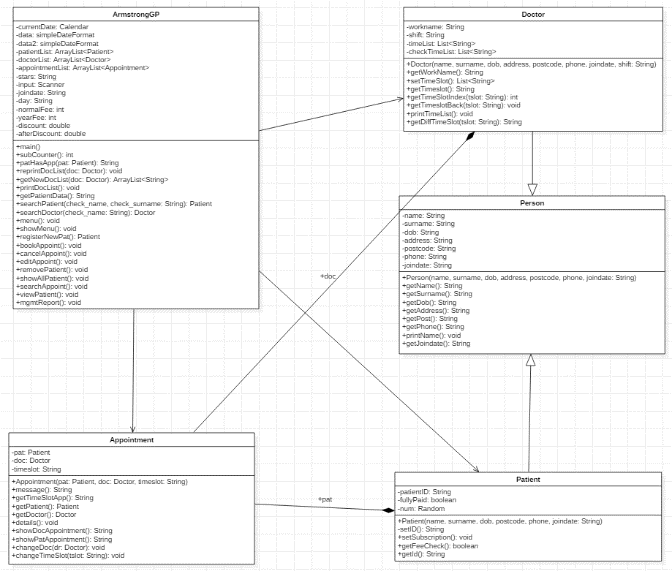
Abstraction is a way to show the user only the important parts of the program and hide the internal details. I java we use abstract classes and methods to achieve this goal. An abstract class is a class which will never directly have object created. It will have common attribute and methods, even abstract methods for their sub or child classes. (Java Tutorial 2019)



# Meeting the design Criteria

Using OOP concepts come in handy when I tried to meet the design criteria. I am very proud of my work there is still some place for improvement, but I learned a lot during the process of creating this application. I started over three times because I didn’t like how the app looked or I thought of doing something different way.

With all the experience I got from this assignment next time I would start differently. I would first design the classes maybe using UML Diagram and then based on that template I would implement the code, it would be easier with a “roadmap”.



In my opinion my application meets the design criteria. The source code responsible for each functionality works given the expected results.

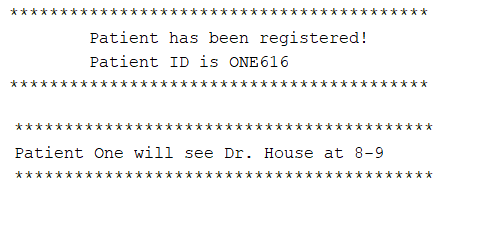
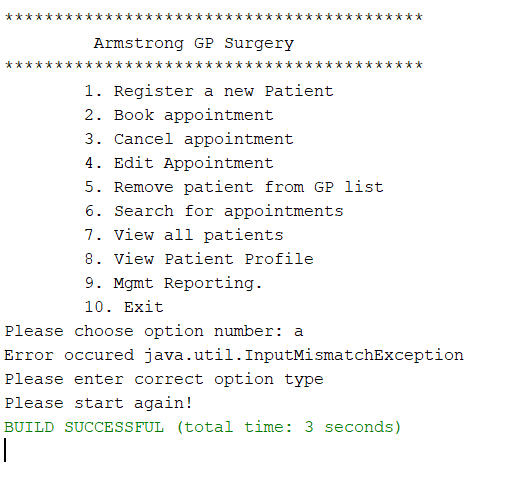


Figure 5 Output of working functionalities

Although the program is prone to errors like entering a character instead if a number as an option that shuts down the program instead of restarting menu again.



Taking in the consideration how much time I spent and the amount of stress experienced I would change the time management scheme and spend more time on research and design, gathering all the concepts I want to use in one place and then into development phase divided into reasonable amount of coding sessions. Another improvement would be making documentation together with application for example describing a method after finishing it.

## Enhancements

In many good programming practices other programmers say that You should revisit your code occasionally just to see that there are things that can be improved. It is called code refactoring. I was trying to follow good coding practices during the creation of the project. (Coding practices 2011) Below I will present the ideas I came up with while thinking of enhancing the program.

### Methods and menu class

While examining my project source code I came up with an idea that I could create a class for Menu and a class for Methods to keep the program more readable. Then in main class I will just have classes instantiation It would look nice and clean.

### Catch statement for edit appointment

Examining my code more I saw tha case 2 for edit appointment function is prone to errors by changing the appointment time to any String:

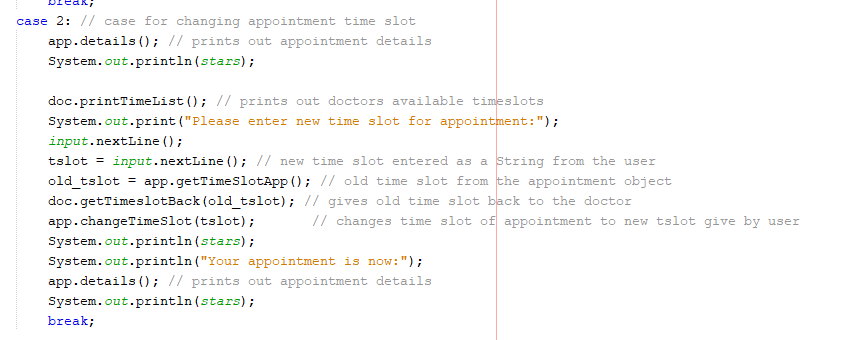


Figure 6

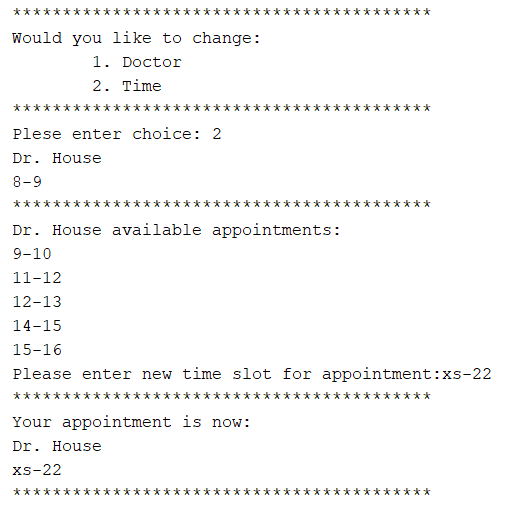


Figure 7

As you can see in Figure 7 the appointment time has been set to “xs-22” which is not a valid appointment time. I could do a switch statement for choosing each appointment so that errors won’t be made.

### Error testing

I definitely could add more try… catch… statements but while trying to implement them on showMenu() method I didn’t get the expected result so many times that I decided I am going back to it later. I could add more if statements for checking that the attributes which are being passed to methods are of correct type or value I just saw that solution on one forum while exploring the information and examples on Polymorphism. These enhancements could make program work smoothly without interruptions.

### Hash maps

Instead of using an Array List for storing the information about appointments I could use hash maps. Hash maps are different from Array List by the fact that Array List are sorted data list and you enter each item on the list using it index. While looking for an element you need to go through every element on the list and compare the attributes until You find that element.

It is not bad If you relatively small lists but going through a list of 50 000 000 could take some time. That’s where hash maps come in handy. Instead of using an int value to represent each object has maps use special “key/value” by which You can find each element. Hash maps don’t care about order of elements it uses hash tables to

#### Advantages

* The purpose of a map is to store items based on a key that can be used to retrieve the item at a later point.
* HashMap is a fail-fast iterator.

#### Disadvantages

* You need to predefine number of elements stored in hash maps
* You can't lock the whole map to find the size of the map. Iterators are weakly consistent.
* You can have a hash collision – this happens when two different keys point to the same cell.
* HashMap does not guarantee that the order of the map will remain constant over time.

# References

1. (Lewis 2008) - Lewis, John; Loftus, William (2008). *Java Software Solutions Foundations of Programming Design 6th ed*. Pearson Education Inc.
2. (W3 Schools) - W3 Schools Java Tutorial, available at : <https://www.w3schools.com/java/java_classes.asp> (accessed 11/04/2019)
3. (Java Tutorial 2019) <https://javatutorial.net/java-abstraction-example>, (accessed 11/04/2019)
4. (Deitel 2015) - Dietel P. and Deitel H.(2015) *Java How to Program Early Objects.* (10th Edition) Essex: Pearson.
5. (Oracle, 2017) - The *Java Tutorials*. Available at: <https://docs.oracle.com/javase/tutorial/java/> (Accessed: 04/04/2019)
6. (Coding practices 2011) - Burak Guzel (2011) Top 15+ Best Practices for Writing Super Readable Code, available at: <https://code.tutsplus.com/tutorials/top-15-best-practices-for-writing-super-readable-code--net-8118> (Accessed 01/03/2019)