**Extract creation class**

**Scenario**

Scrum meeting- Chief of police department

The NT police department is currently ongoing a state of high criminal activities and therefore, require more police cars than we currently have. They require a system which will be able to provide details on the police car factory and feature and details of their vehicle.

**Scrum meeting discussion points**

Product demonstration to the client (check git)

Retrospective and discussion of next sprint planning

Next sprint planning (Product owner will prioritise the next two task)

Burn down chart on the number of tasks done and how many remains vs time - possible but not applicable for this assignment

**Grooming**

Refactoring technique: Extract creation class

Reasoning/motivation:

Classes are created and given specific responsibilities and each class as a primary responsibility which is object creation. These classes are easy and clear to understand however, as the program expands these classes start dominating the public interface. As a result, some classes are performing more responsibilities than intended. The solution for this is to create another class with the intent to communicate the act of object creation by creating instance of object in special classes.

Benefits

By using the Extract creation class refactoring technique it helps:

* maintain adherence
* highlights the Single Responsibility Principle
* clear and maintainable code
* Single-responsibility classes are more reliable
* Tolerant of changes (agile principle)

Disadvantage

Extensive use of the Extract Creation class refactoring technique can resort to the use of the inline class.

However, if not refactored the system can lead the explosion of class, which can make a system not simple to work with at all and not adaptable to change.

**UML diagram**

Example

|  |
| --- |
| **PoliceCarController** |
| Police Car |
| Get **PoliceCarfactory** |

|  |
| --- |
| **PoliceCarfactory** |
| policeCar  ServiceRequired  Mileage |
| return policeCar |

|  |
| --- |
| **PoliceCarController** |
| policeCar  ServiceRequired  Mileage |
| return policeCar |

Risk of refactoring: Medium

The system is still functional and can still be utilise however, the system development is based on agile principles and values. Hence, agile development encourages changes and if the system is compress it is extremely difficult to add or edit code in the next iteration of the system

**Mechanics**

1. Identify a class that is overloaded with responsibility and need Creation Methods. And determine what to extract

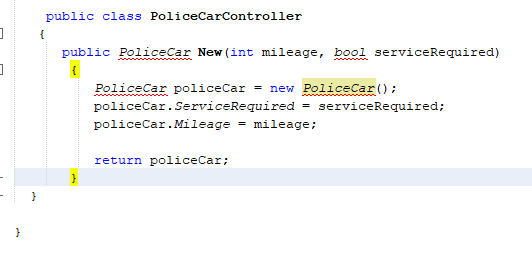
2. Create a Creation Class. Name it purposely, this class contains the extracted operations and link old to new class.

3. Move all Creational Methods from original class to your new class, making sure that all protection privileges are accounted for.

4. Change all callers to obtain object references from your new Creation Class, rather than

**Before and After**

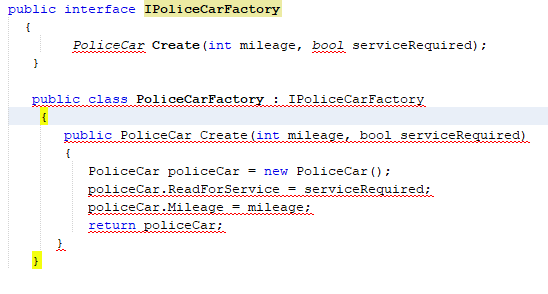
The police car controller will class will become over populated with responsibilities and the system can become quite complex.

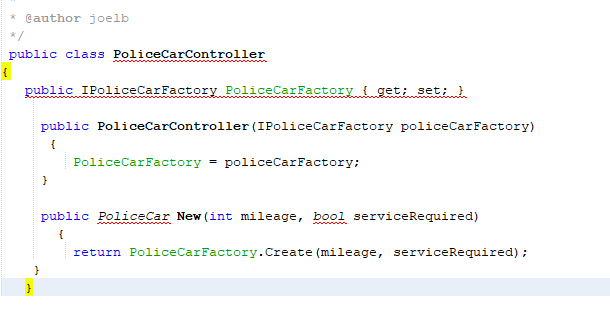


After

The current code works fine in a small system however, when system become complex and we need to create multiple police car with unique details then it is problematic. The solution for this is to create a class which will solely focus on creating instances.

Hence: the controller no longer needs to focus on creating police cars instead we now create a class with the distinct purpose of creating factory police cars.





Risk : The risk is Low as this is a system on police car factory

**Testing**

Iterative testing method is important when refactoring a medical system. Refactor small bits at a time and maintaining the system functionality. This ensure that we utilise continuous integration and the application is always ready for use a deployment. However, the are several components that can be tested through unit testing. Component such as the policeCarfactory method can be tested.

Further information on the refactoring method, reasoning, risk and more

as per document on attachment.