



MONASH University

# EXTRA CREDIT SECTION

Kaihan Xie 28560949  
Wenchu Du 29030250

In the first section, we decided to create an attribute named 'family' in Rocket class to achieve the goal, because we thought that the family information only contained the name of the family. Before implementing the field, we created several test cases for that, which included:

1. The family attribute should not be empty or null. This is the basic validation because family is the vital information for rocket series.
2. The family should be capitalized on each word. Because it is the name of a series of rockets, it should be capitalized to illustrate its importance.
3. The family should be unique. Because family name is used to represent a series of aerospace projects, thus it should be unique as an identity.
4. For the name variable of the rocket, we assumed that it should contain the family name. We searched online and found that, for almost every rocket, it has a family name and always use it as a prefix of the rocket name.
5. After we created test cases, we create the family variable and developed the unit testing. We ran the unit test at first and coding the validation part which the unit test is failed. We do this step iteratively until we pass all the unit test.

In the second section, we decided to create a new class named Payload to the domain model and change the type of attribute payload in the Rocket class to have the capacity to bear objects of Payload. Moreover, in the Payload class, we assumed that there are several attributes to achieve this class, which included: name, category, country, manufacturer, function and mass. In this section, we used Rocket class as a reference, because the payload is also an item which should have a nationality and its own manufacturer to deal with the specific task. It needs to be mentioned that mass is a vital attribute of payload because, for each rocket, its capacity of the payload is fixed for every orbit, the mass should be strictly below or equal to the payload mass restriction. Before we implement the code of payload class, we design several test cases to validate the payload class, which included:

1. Name of payload should not empty or null. The name is an identity to identify the payload from others.
2. The name should be unique. The reason is the same as above.
3. The category should not be empty or null. The category can be spacecraft, probe or other devices, which should classify payload into several different areas.
4. The category should be unique. Because it is used to classification, the name of the category should be different from each other.
5. The country should not be empty or null. The country is used to identify the nationality of the payload. It is usually different from the nationality of the rocket. For instance, in a news recently, due to the short of the fund, NASA would use rocket made by Russia to carry their experimental devices in space. Thus, we assumed that the country of payload should be different from the country of the rocket.
6. The country should be capitalized on each word. This is common sense.
7. The manufacturer should not be empty or null. Each payload has its own manufacturer.
8. The function should not be empty or null. The function is used to identify the purpose of the payload. For instance, there is a specific robot is used to find the water source from Mars.
9. Mass should not be empty or null. As we discussed before, the mass of payload is one of the most important attributes which determines whether or not the rocket has the ability to carry this specific payload.

After we created test cases, we developed the unit test to validation. Then, as followed the process of TDD, we implied the code which includes creating the Payload class and change the type of payloads attribute in Rocket class. When finished the implementation of extension and modification of domain model, we ran all unit test to validate. After running the unit test, we refracted the coding iteratively until all test units have been passed.