turntabl

Java - Inheritance & Polymorphism

Exercise 01 - Group Work

A fleet of spaceships require a seat booking system for a passenger to book a seat on a flight. We need to provide them with the ability to book!

Humans can book ‘human’ seats and martians can book ‘martian’ seats.

1. There should be one seat booking system, which all spaceships can use.

2. Implement classes for both these seat types. Both of these classes should inherit from a parent **Seat class** and have an **ID** and the **name** of the person booking. Think about where these variables should be declared.

3. We need all the spaceships to hold a structure of **Seats**. Think about where this structure should be declared.

4. Create a function for the spaceship classes, which allows a seat to be ‘booked’ i.e. a Seat object is filled with passenger data. Think about the function inputs required to make an appropriate seat booking, and how to ensure the structure does not go over capacity.

5. Passengers get a meal on the flight. The meal is **different** for humans and for martians, but everyone gets one. Using method **overriding**, print out the meal served for that seat, when it gets booked.

6. Write unit tests to ensure that the seat type is correct when booked, and that you cannot book at capacity.

7. Humans believe Pluto is a Moon, Martians believe Pluto is a Planet. How can we implement Pluto to best accommodate both?

8. Don’t forget unit tests!

Exercise 02 - Code Reviews

Swap code with someone else. Check each other's code and provide feedback on

- Correct relationships between classes
- Clean code
- Test the booking functionality!

Exercise 03 - Quiz

Multiple-choice questions on Method Overloading & Method Overriding - Complete here

Exercise 04 - STRETCH Work

Have you 100%'d the Polymorphism quiz?
Have you finished all features, refactored and tested?
Have you reviewed each other's code?

Here's some stretch work to do to ensure you've really grasped the concepts from today:

a) **Research:** Static/dynamic binding. Why is this relevant to Polymorphism?

b) **Question:** Can *final* classes be inherited? Why/why not?

c) **Question:** Can *static* methods be overridden? Why/why not?