

Horse Racing Simulator Report Benjamin Maltby 230367278

Encapsulation Explanation

Encapsulation is used in the Horse class so that the private data such as the horse's name, symbol, distance travelled, fallen status, and confidence is protected from outside access to the class.

Accessor (getter) methods:

- getName()
- getSymbol()
- getDistanceTravelled()
- hasFallen()
- getConfidence()

Mutator (setter) methods:

- setSymbol(char newSymbol)
- setConfidence(double newConfidence)
- fall()
- moveForward()
- goBackToStart()

These methods make sure that the data can't be accessed or changed unless they're done through the interface of these methods. Also, I used validation like clamping the confidence level between 0 and 1 when the setConfidence method is called which ensures data integrity.

Testing Evidence

Test 1: moveForward()

- Created a horse and called moveForward(), checked distance increased and it did.

Test 2: fall()

- Called fall(), confirmed horse has fallen and the symbol changed to an X.

Test 3: Confidence bounds

- Set confidence above 1.0 and below 0.0 to make sure the clamping calculation is correct.

Horse Racing Simulator Report Benjamin Maltby 230367278

Test 4: Adjustments after win/fall

- Verified winning slightly increases confidence by 0.1.
- Verified falling decreases confidence by 0.1.

Tests confirmed correct behaviour of movement, falling, and confidence altering.

Identified Issues and Fixes

Issue 1: Race only ended when horse won

- Fixed by checking if all horses had fallen and stopping the race if they had.

Issue 2: Fixed number of lanes (3)

- Updated Race class to allow any number of lanes dynamically, which can be set by the user on start.

Issue 3: Crashing on empty lanes

- Added null checks throughout race logic, so the program isn't attempting to alter a horse's logic when the horse is a null pointer.

Issue 4: No setup from terminal

- Created Main.java to allow race setup interactively via command-line prompts for a smooth user experience.

Updated Code Explanation

Race class now uses List<Horse> lanes.

- Constructor initializes the list with empty lanes.
- startRace() allows starting with empty lanes.
- moveHorse(), printLane(), and all race logic safely handle null horses.
- Confidence is automatically adjusted after winning or falling.

Main class includes function testing and full user prompt for race setup.