Solid violation in returnBike functionality

1. Single responsibility
   Each function only serves 1 purpose
2. Open/closed principle
   No extra modification to existing function. Functionality of each class is already separated.

3. Liskov Substitution Principle

   There is no inheritance of significant meaning ( only have inheritance from baseController and baseScreenHandler, both does not provide any attributes or method).
4. Interface Segregation Principle
   There is no use for interface in this implementation because there is only one flow to take. The most probable implementation could be for calculateRent to be different with each type of vehicle rented, but that approach is not used in the project.
5. Dependency inversion Principle
   Refund needs to touch implement-specific details ( refund if moneyReturn >0). This can be avoided by making a shouldRefund that return the result.