# EXCEPTION AND PITFALL INSPECTION

**Team 3:** D-Encoders

**Project's name:** Ferrinventory

**Inspection by team 2:** BugBusters

**Code version:** 0.4

**Date:** Wednesday, june 21th, 2023

**INDICATOR 1: Exceptions**

**Aspects to corroborate:**

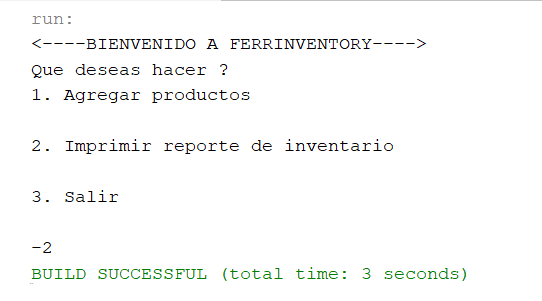
* **Problem:** Check with letters

**Consequence:** The program stops running because it has an error.

**Evidence: **

* **Problem:** Check with negative numbers.

**Consequence:** The program doesn’t perform any action.

**Evidence:** 

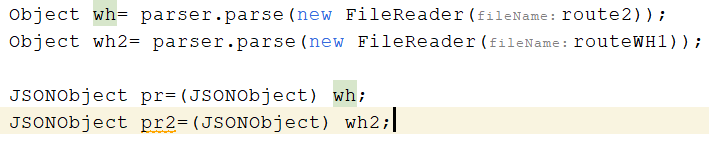
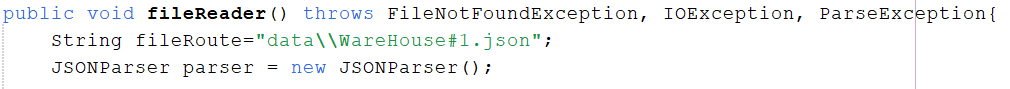
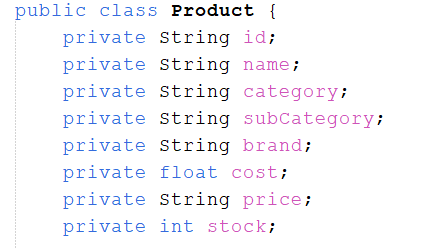
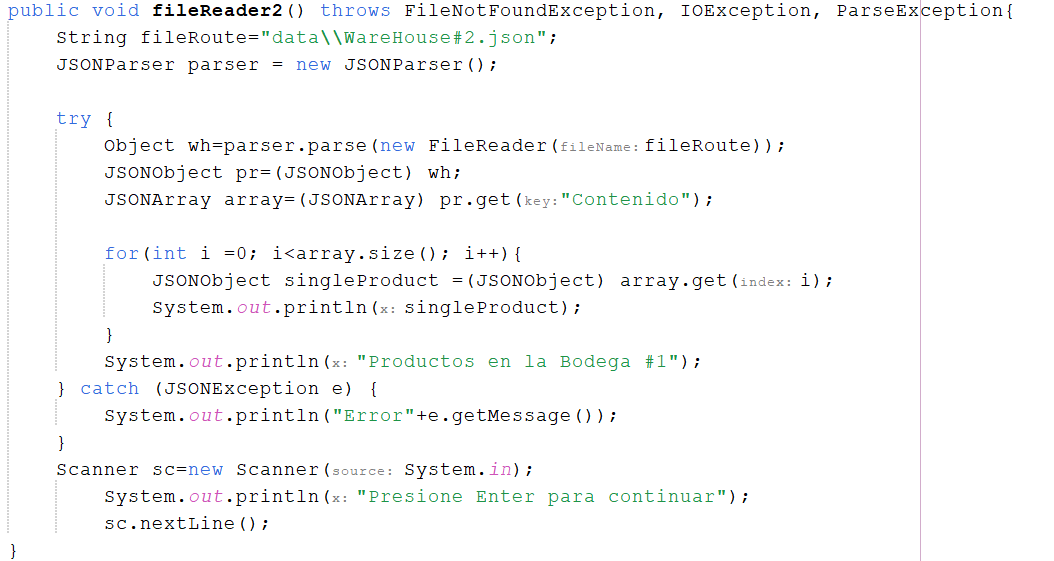
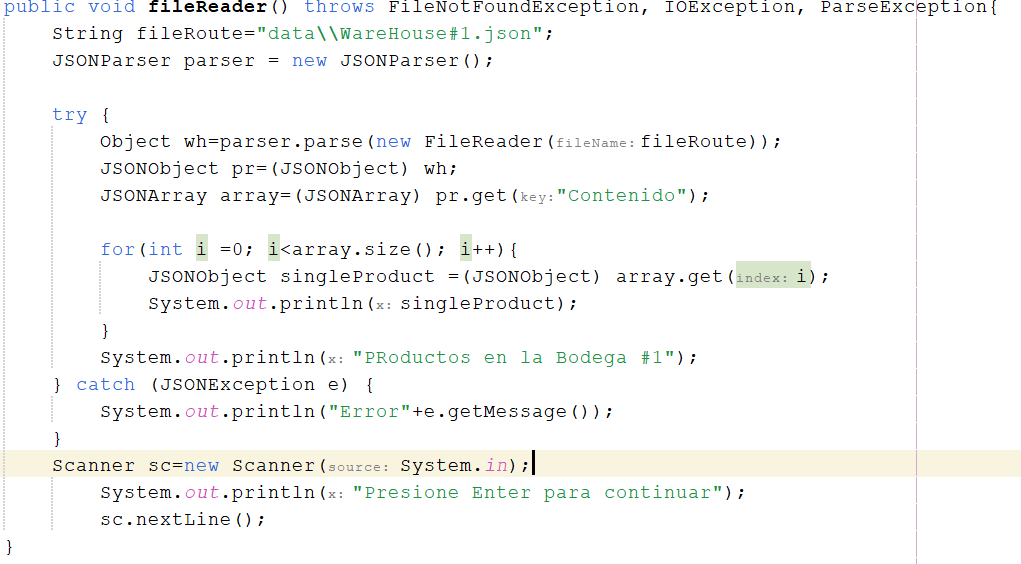
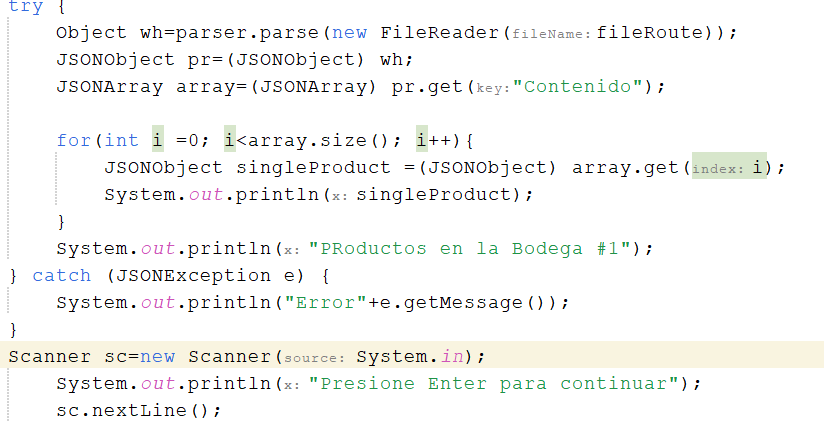
* **Problem:** Check large numbers

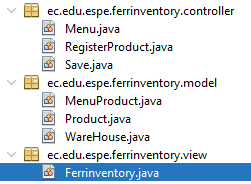
**Consequence:** The program stops running because it has an error.

**Evidence:**

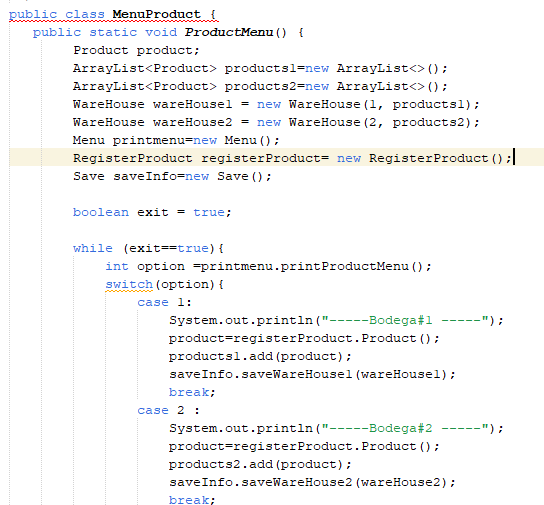
**INDICATOR 2: Pitfalls**

**Aspects to corroborate:**

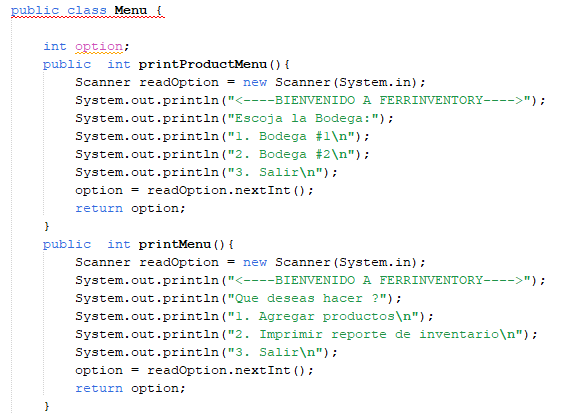
* **Uncommunicative Names (“wh”, “wh2” doesn’t communicate that it represents)**
* **Inconsistent Names (te method name is fileReader and the name of the String is filerRoute)**
* **Types Embedded in Names(variables are clean)**
* **Long Methods(the methods do the same but with some changes so there are unnecessary lines of code.)**
* **Duplicate Code(In the save class there are 4 methods in which the code is copied and small changes are added.)**
* **Long Message Chains(messages are short)**
* **Class Explosion (Doesn`t have subclasses that does the same thing)**

****

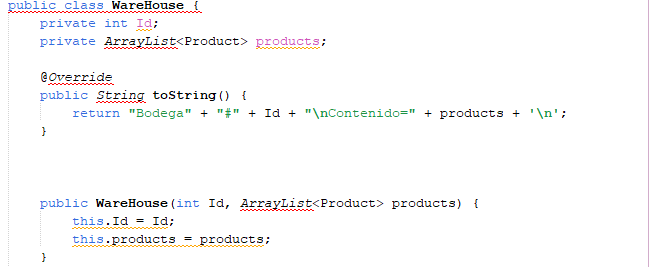
* **Large Message Chains (No more than 1 call, everything ok)**

****

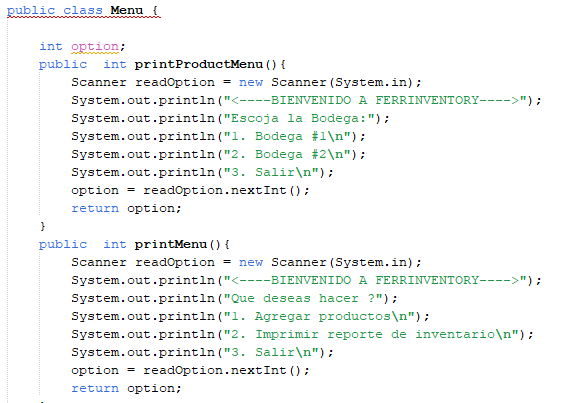
* **Large Classes (The project has max. 2 methods per class)**

****

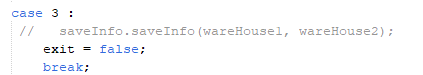
* **Conditional Complexity (Not hard to understand what the system is doing)**

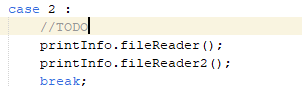
****

* **Oddball Solution (Maybe similar methods but not the same functionality)**

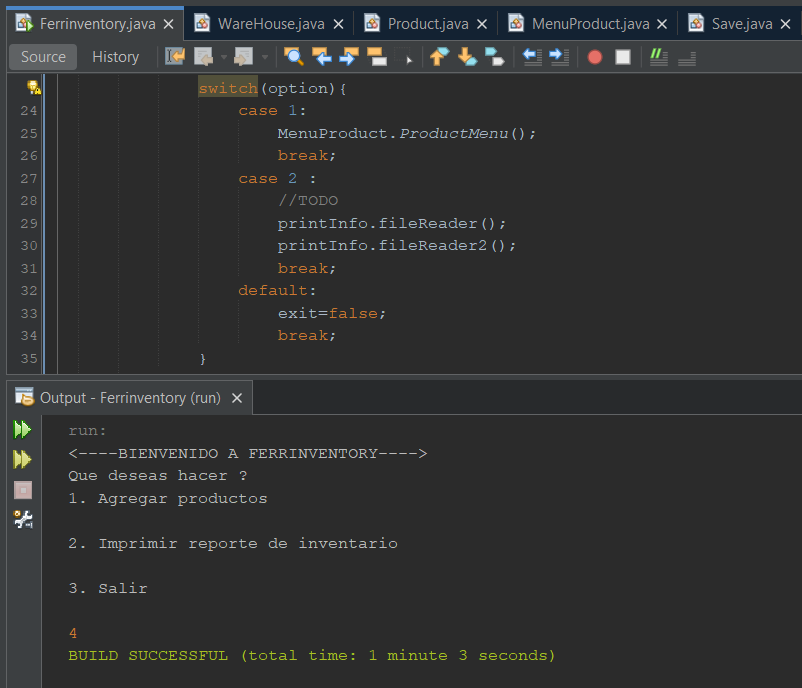
****

* **Redundant comments**

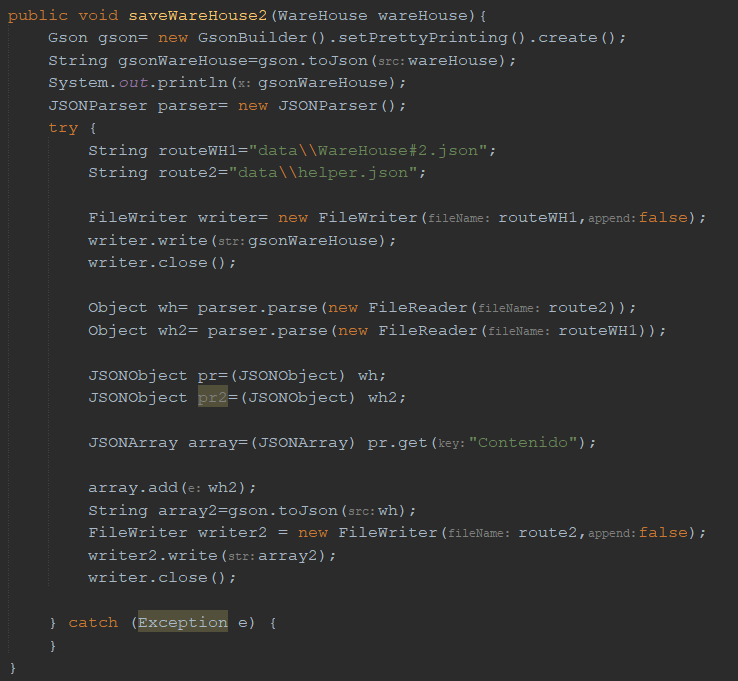
****

****

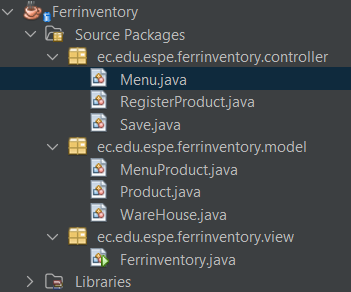
* **Dead Code (case 3 is not used even though it is printed in menu)**

****

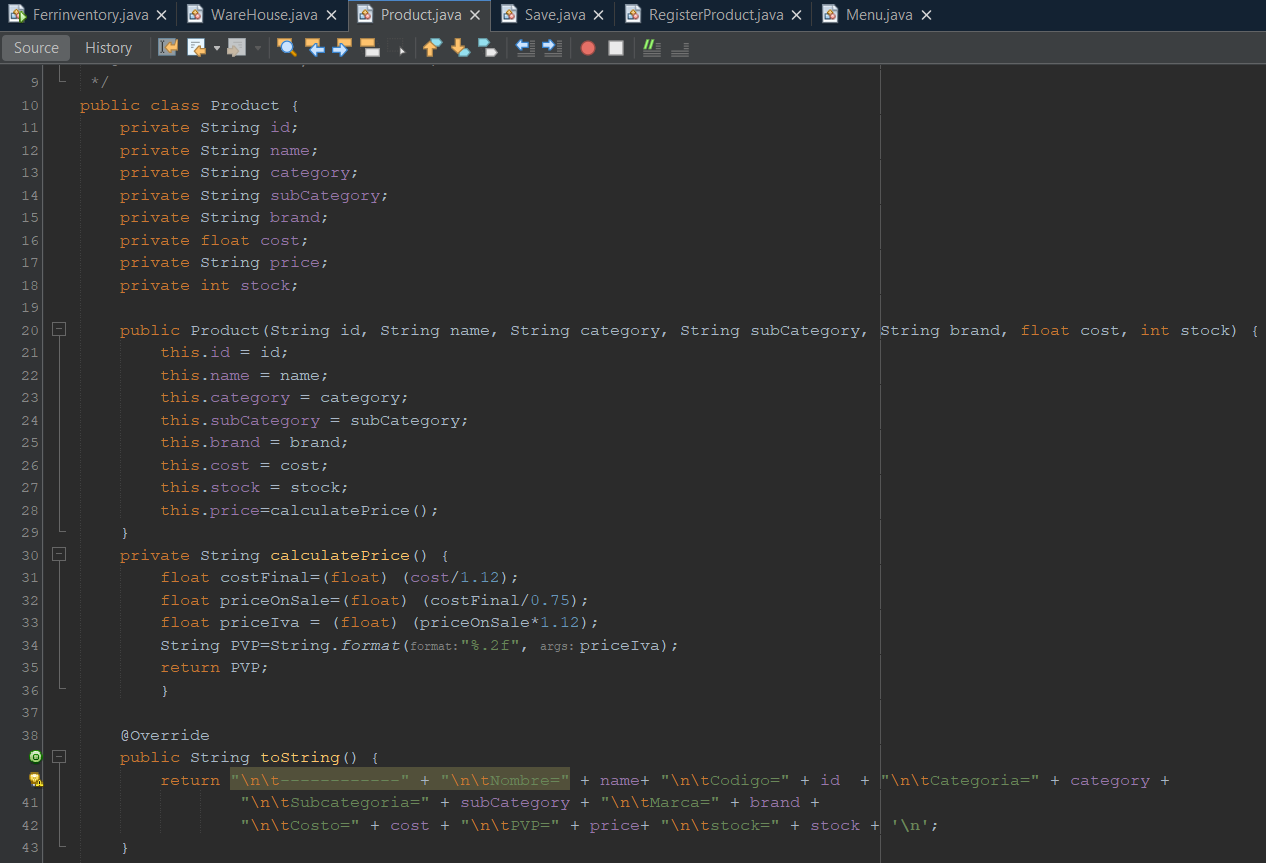
**(the exception isn’t used)**

****

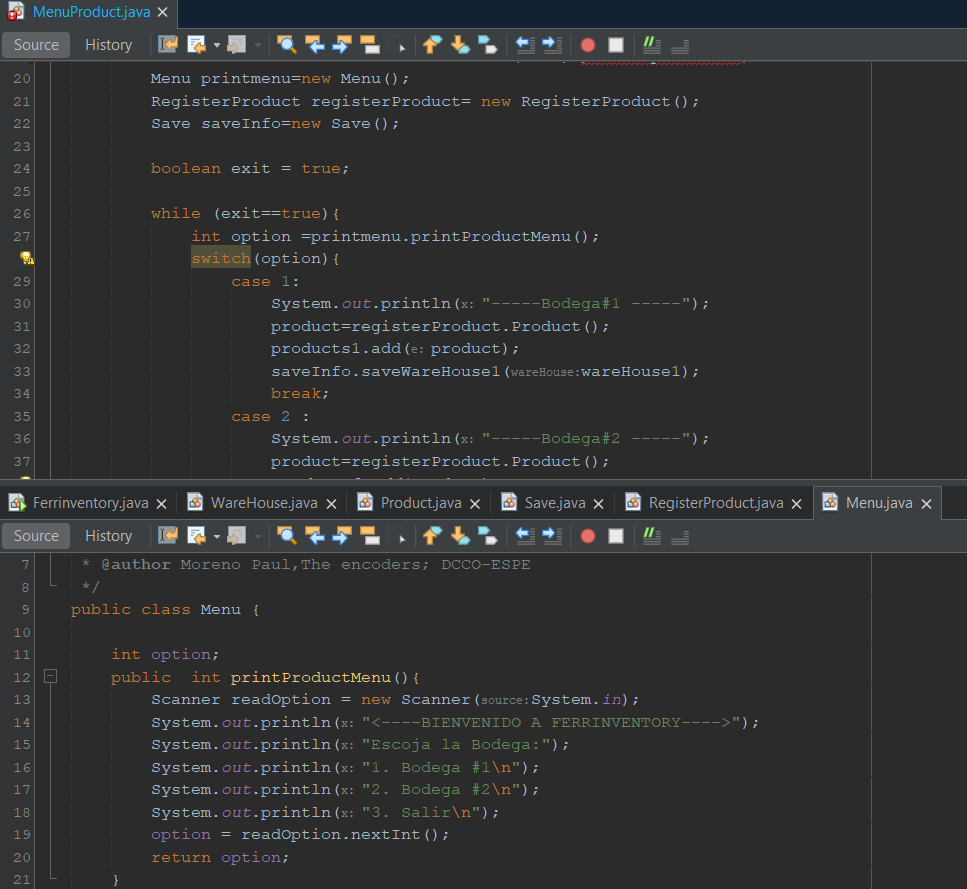
* **Speculative Generality (menus could be in the same controller package, also, use the default domain of the university and not the team’s domain)**

****

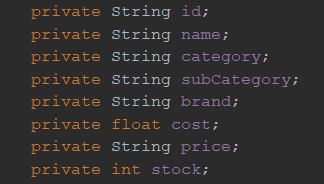
* **Temporal Field (ok, all variables are used)**

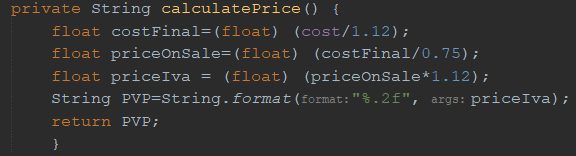
****

* **Refused Bequest (menu class only print the message when printProductMenu method could be in the same MenuProduct class)**

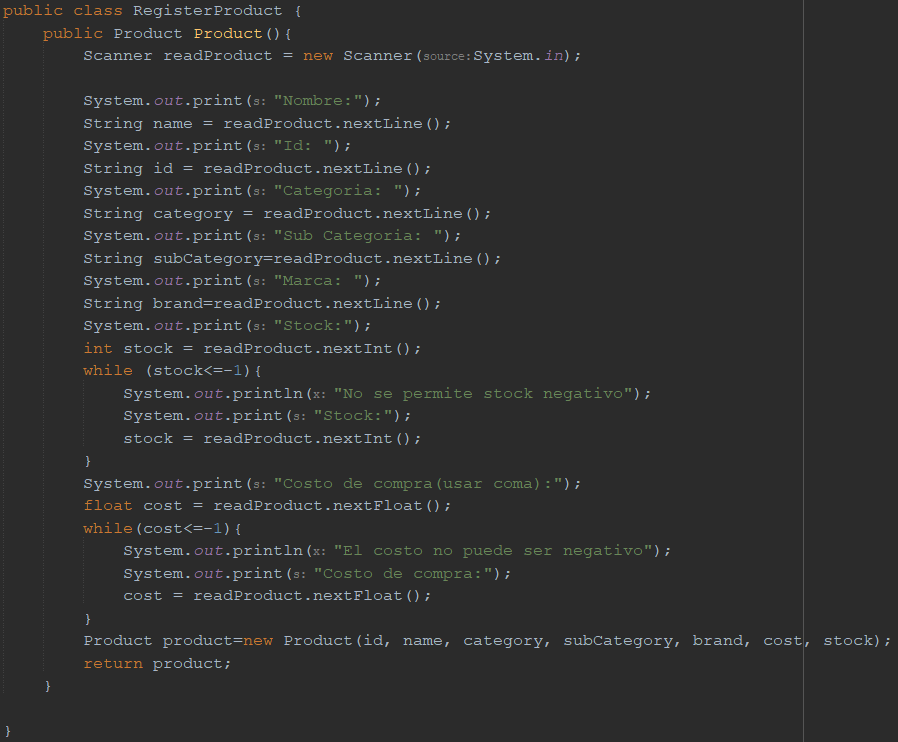
****

* **Inappropriate Intimacy (ok, internal fields and methods was encapsulated for only one class)**

****

****

* **Feature Envy (ok, the RegisterProduct class accesses the same number of data from the public Product method)**

****