

FILE User.cs

1. What bad practices or “code smells” can be identified on the code that may difficult its analysis or maintenance?

1. Non conventional field names
2. Nested classes
3. No encapsulation

2. What is your proposal for an easier to maintain and more simpler code?

1. Added use of constructors for each class contained in the file
2. Changed public fields to properties for all classes to enhance access control
3. Improved encapsulation for a more modular and reusable code

Please note that this can be enhanced further if not because of the Data.cs file which is required not to modify.

FILE Importer.cs

1. What bad practices or “code smells” can be identified on the code that may difficult its analysis or maintenance?

1. Unnecessary use of dynamic to handle different object types
2. No matching pattern use in case ThePerson type is invalid
3. No error handling in case ThePerson is invalid

2. What is your proposal for an easier to maintain and more simpler code?

1. The use of a matching pattern to solve different object types
2. The use of static using for person to access the static members of the person class
3. Error handling in case ThePerson is invalid

3. Is there any logical error that may occur on the code at run time under very specific circumstances?

Yes , for the case that the provided Id's are incorrect, the execution will just go with the basic person and print it as many times as ids are provided regardless of the current id validity.

Also the detected error handling issue reported above may cause a run time error