Answers to the questions:

1. What is the principle of sole responsibility? What's its purpose?

2. What features do you have, in your opinion, a "good" code or clean code?

Ans1.Single responsibility principle stresses that a class should only cater to the implementation of a single feature or functionality in the software(i.e there should be loose coupling)

Ans2.A good piece of software is well segregated into the layers of essential modules for instance view, model, presenter and viewmodel so that it is easy to understand, modify, debug and evolve. Other than that A well written software should be adher to SOLID principles of software design i.e. (1)A class should only implement single feature or responsibility, (2)it should be closed for modification and open for extension to prevent spaghetti code, (3)object should be replaceable with their subtypes without the need for a extra conditional logic(for instance in vehicle class both truck and car have their own accelerate function by the virtue of polymorphism) (4)A developer should not be bound to implement single of a few large interfaces if he does not need them and many client specific interfaces are better than one general purpose interface and lastly (5)There should be dependency inversion which means highlevel modules should not be depended upon the lower level modules. Higher level modules classes should depend upon abstractions and protocols and not on concrete classes