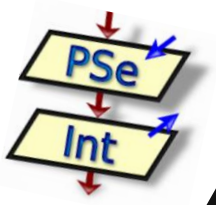
UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

OBJECT ORIENTED PROGRAMMING



Student:	Llumiquinga Moreno Jerson.
E-mail:	jsllumiquinga1@espe.edu.ec
#Cell:	0980460057
Professor:	Edison Lascano.
E-mail:	jelascano@espe.edu.ec
#Cell:	0961195050
NRC:	14539

MAY24 - SEP24

Single Responsibility Principle (SRP):

This principle establishes that a class must have a single reason for changing, that is, it must only have one responsibility. In our code, the StudentToMongo class handles both the connection to MongoDB and the business logic related to obtaining and storing students. This can be considered a violation of the SRP, since the class is handling two different responsibilities: database connection and business logic.

Open/Closed Principle (OCP):

This principle states that classes should be open for extension, but closed for modification. In our case, if you needed to change the way a student is saved or retrieved from the database, you would need to modify StudentToMongo. This suggests a lack of adherence to the OCP.

Dependency Inversion Principle (DIP):

This principle states that high-level modules should not depend on low-level modules; both must depend on abstractions. Currently, StudentToMongo depends directly on the MongoDB implementation, making it difficult to change the database or storage form without modifying the code.

Single Responsibility Principle (SRP):

Detected Issue: The GradeCalculator class currently has multiple responsibilities:

Calculate the average of the grades.

Determine the status (approved/failed) and update a graphical interface component (JLabel).

Validate the notes entered in the text fields of the graphical interface (JTextField).

This violates SRP because the class mixes business logic (average and state calculation) with presentation logic (GUI component manipulation and input validation).

Open/Closed Principle (OCP):

The determineStatus and validateGrades logic is coupled to the graphical interface. If you wanted to change how the interface is handled or add new validation criteria, you would have to modify the GradeCalculator class.

Dependency Inversion Principle (DIP):

The GradeCalculator class depends directly on specific GUI components (JLabel, JTextField, JOptionPane), making it difficult to reuse the calculation and validation logic in a non-graphical environment or with different interfaces.