ASSIGNMENT 2- Design and Implementation of Class, Function, and Model in OpenModelica

---

Objective

To understand and implement:

- Basic Class structure
- Function definition
- Use of a function inside a model
- Simulation of a simple physical system

Background

In Modelica:

- A class defines structure and reusable components.
- A function performs mathematical computation.
- A model represents a dynamic system that can be simulated.

This assignment will test your understanding of these three elements and their integration.

Problem Statement

Design a simple physical system representing vertical motion under gravity and compute its total mechanical energy using a function.

You must:

1. Define a class to store system parameters.
2. Define a function to compute total energy.
3. Create a model that uses both.
4. Simulate and analyze the results.

Tasks to Perform

Part A: Create a Parameter Class

1. Create a new Modelica class named SystemParameters.
2. Inside the class:
    - Define mass of the object.
    - Define gravitational acceleration.
    - Define initial velocity.
3. Assign appropriate default values.
4. Ensure the class contains only parameter declarations.
5. Do not include equations or algorithms in this class.

Deliverable:
A reusable parameter container class.

Part B: Define a Function

1. Create a function named TotalEnergy.
2. The function must:
    - Accept mass, velocity, and height as inputs.
    - Return total mechanical energy as output.
3. The function must:
    - Use an algorithm section.
    - Compute:
        - Kinetic Energy
        - Potential Energy
        - Total Energy
4. Ensure:
    - No derivatives are used.
    - No equation section is used.

Deliverable:
A computational function that calculates total mechanical energy.

Part C: Create the Main Model

1. Create a model named VerticalMotionSystem.
2. Inside the model:
    - Declare variables for velocity and height.
    - Declare a variable for total energy.

3. Instantiate or reference the parameter class.
4. Write differential equations for:
    - Velocity under gravity.
    - Height change based on velocity.
5. Call the TotalEnergy function inside the equation section.
6. Assign the computed value to the total energy variable.

Deliverable:
A complete dynamic model that uses both the class and function.

Part D: Simulation

1. Simulate the model for 10 seconds.
2. Plot the following variables:
    - Velocity
    - Height
    - Total Energy
3. Observe:
    - How velocity changes over time.
    - How height varies.
    - Whether total mechanical energy remains constant.