```java

/**
 * Describes a module, contains its code, name and the weights of its controlled assesments.
 */
public class ModuleDescriptor {

    private String code;

    private String name;

    private double[] ContinuousAssignmentWeights;

    /**
     * ModuleDescriptor constructor
     *
     * @param code
     * @param name
     * @param ContinuousAssignmentWeights, each weight cannot be negative and weights must sum to 1
     */
    public ModuleDescriptor(String code, String name, double[] ContinuousAssignmentWeights) {

        // Validates that paramaters are not empty
        if (code == "") {
            throw new IllegalArgumentException("Code cannot be empty");
        } else if (name == "") {
            throw new IllegalArgumentException("Name cannot be empty");
        } else if (ContinuousAssignmentWeights.length == 0) {
            throw new IllegalArgumentException("Weights cannot be empty");
        }

        // Validates the ContinuousAssignmentWeights sum to 1
        double weight_sum = 0.0;
        for (double weight : ContinuousAssignmentWeights) {
            if (weight <= 0) {
                throw new IllegalArgumentException("Weight cannot be less than or equal to 0");
            } else {
                weight_sum += weight;
            }
        }
        if (weight_sum != 1) {
            throw new IllegalArgumentException("Weights need to add to 1");
        }

        // Initializes variables
        this.code = code;
        this.name = name;
        this.ContinuousAssignmentWeights = ContinuousAssignmentWeights;
    }

    public String getModuleCode() {
        return code;
    }
}

```