**Các chuẩn thiết kế lớp gồm:**

**Luôn để phần dữ liệu private**

* Đây là điều đầu tiên và quan trọng nhất để đảm bảo tuân thủ tính đóng gói dữ liệu.
* Dữ liệu có thể thay đổi thường xuyên nhưng cách mà dữ liệu đó được sử dụng thường ít thay đổi hơn.
* Ta nên phân tách rạch ròi ra thành hai phần: các dữ liệu của lớp luôn để private và muốn truy cập chúng từ bên ngoài lớp ta cung cấp các getter/setter tương ứng.
* Điều này vừa đảm bảo tính an toàn cho dữ liệu của lớp vừa hạn chế các lỗi có thể xảy ra.
* Ví dụ:

public class Pet {

    private String name; *// tên*

private float age; *// tuổi*

*// các getter/setter:*

public String getName() {

        return name;

    }

    public void setName(String name) {

        this.name = name;

    }

    public float getAge() {

        return age;

    }

    public void setAge(float age) {

        this.age = age;

    }

}

**Luôn khởi tạo dữ liệu**

* Java không tự động khởi tạo giá trị mặc định cho các biến cục bộ. Do đó với các biến cục bộ bạn phải chủ động khởi tạo hoặc gán giá trị trước khi sử dụng.
* Đối với các thuộc tính, Java sẽ khởi tạo các giá trị mặc định của kiểu tương ứng. Tuy nhiên bạn không nên phụ thuộc vào điều này mà nên chủ động khởi tạo giá trị mong muốn cho các thuộc tính trong các phương thức khởi tạo.
* Điều này sẽ tạo thói quen tốt và tránh được ngoại lệ NullPointerException.
* Ví dụ:

public class Pet {

    private String name; *// tên*

private float age; *// tuổi*

private float weight; *// cân nặng*

*// khởi tạo giá trị cho các thuộc tính trong constructor:*

public Pet() {

        age = 0;

        name = "";

        weight = 0;

    }

*// các getter/setter:*

public String getName() {

        return name;

    }

    public void setName(String name) {

        this.name = name;

    }

    public float getAge() {

        return age;

    }

    public void setAge(float age) {

        this.age = age;

    }

}

**Không sử dụng quá nhiều kiểu nguyên thủy trong 1 lớp**

* Để thực hiện ý tưởng này ta gom nhóm các thuộc tính liên quan đến nhau lại thành một lớp nhằm đơn giản hóa và tiện quản lý.
* Ví dụ:

public class Customer {*// thông tin về họ, đệm tên có thể thay thế không?*    private String firstName;

    private String lastName;

    private String midName;

    private String email;

    private int age;

    private String phoneNumber;

*//...*

} *// đổi thành:*public class Customer {

    private FullName fullName;

    private String email;

    private int age;

    private String phoneNumber;

*//...*

}

*// tách lớp FullName để lưu trữ các thông tin về tên*

class FullName {

    private String firstName;

    private String lastName;

    private String midName;

*//...*

}

**Không phải tất cả các thuộc tính đều cần có getter/setter**

* Không phải tất cả các thông tin trong một lớp đều cần có getter hoặc setter. Ví dụ xét thông tin nhân viên chẳng hạn, bạn không cần thay đổi ngày vào làm việc tại công ty sau khi thông tin đó được gán. Vì đơn giản thông tin đó là cố định. Vậy ta chỉ cần cung cấp getter cho thông tin này.
* Ví dụ khác: thông tin nhạy cảm về số điện thoại, địa chỉ của khách hàng đặc biệt không muốn tiết lộ ra ngoài chẳng hạn. Lúc này ta không cần cung cấp các phương thức getter/setter cho các thông tin này nữa.

**Chia nhỏ lớp thực hiện quá nhiều nhiệm vụ**

* Nếu một lớp nhiều thông tin phức tạp ta nên chia nhỏ lớp đó ra để thuận tiện quản lý và sử dụng. Nhưng điều đó không có nghĩa là ta chia nhỏ đến mức mỗi lớp chỉ có 1 thuộc tính hay 1 phương thức.
* Chúng ta chia nhỏ lớp sao cho mỗi lớp chỉ thực hiện vừa đủ những thứ cần thiết liên quan.
* Chẳng hạn lớp sinh viên chứa các thông tin sinh viên và môn học, lớp học, điểm cùng với các phương thức tính toán điểm, xếp hạng học lực,.. thì ta chia lớp đó thành các lớp độc lập: sinh viên, môn học, lớp học.
* Cụ thể:

*/\*\**

*\* lớp Student để mô tả thông tin sinh viên*

*\*/*

public class Student {

*// các thuộc tính:*

private String studentId;   *// mã sinh viên*

private String fullName;    *// họ tên đầy đủ*

private String email;       *// địa chỉ email*

private String address;     *// địa chỉ cư trú*

*// thông tin liên quan môn học:*

private String subjectId;   *// mã môn*

private String subjectName; *// tên môn*

private int numOfLesson;    *// số tiết học*

private int credit;         *// số tín chỉ*

*// thông tin liên quan khóa học học:*

private String courseId;     *// mã lớp*

private float gradeLevel1;  *// điểm hệ số 1*

private float gradeLevel2;  *// điểm hệ số 2*

private float avgGrade;     *// điểm TB*

private String courseName;   *// tên lớp*

private String instructor;  *// tên giảng viên*

*// các phương thức:*

public void doHomework(){

*//...*

}

    public void addNewSubject(String id, String name){

*//...*

}

    public String calculateAvgGrade(){

*//...*

}

    public String studentLevel(float grade) {

*//...*

}

}

Chuyển thành:

*/\*\**

*\* lớp Student để mô tả thông tin sinh viên*

*\*/*

public class Student {

*// các thuộc tính:*

private String studentId;   *// mã sinh viên*

private String fullName;    *// họ tên đầy đủ*

private String email;       *// địa chỉ email*

private String address;     *// địa chỉ cư trú*

*// các phương thức:*

public void doHomework(){

*//...*

}

*//...*

}

*// lớp mô tả thông tin liên quan môn học*

class Subject{

    private String subjectId;   *// mã môn*

private String subjectName; *// tên môn*

private int numOfLesson;    *// số tiết học*

private int credit;         *// số tín chỉ*

public void addNewSubject(String id, String name){

*//...*

}

*// ...*

}

*// lớp mô tả thông tin khóa học*

class Course {

    private Student student;    *// thông tin sv*

private Subject subject;    *// thông tin môn học*

private String courseId;     *// mã lớp*

private float gradeLevel1;  *// điểm hệ số 1*

private float gradeLevel2;  *// điểm hệ số 2*

private float avgGrade;     *// điểm TB*

private String courseName;   *// tên lớp*

private String instructor;  *// tên giảng viên*

public String calculateAvgGrade(){

*//...*

}

    public String studentLevel(float grade) {

*//...*

}

*//...*

}

**Đặt tên của lớp, thuộc tính và phương thức sao cho phản ánh được nhiệm vụ của chúng**

* Phần này đã trình bày chi tiết trong từng bài học, trong đó: tên của biến, thuộc tính đặt sao cho thể hiện ý nghĩa mà biến mô tả. Tên thuộc tính là danh từ, cụm danh từ, tính từ, trạng từ kết hợp danh từ.
* Với tên lớp cũng vậy. Tên lớp luôn là danh từ, cụm danh từ hoặc tính từ kết hợp danh từ. Ví dụ: Student, GraduatedStudent, BillingAddress, LearingProcess, ProcessingOrder, CompletedOrder…
* Với tên phương thức thì luôn thể hiện hành động, nhiệm vụ mà phương thức đang thực hiện. Do đó tên phương thức là động từ. Chỉ trừ một số lớp có nghiệp vụ quá đặc thù thì đặt tên phương thức theo nghiệp vụ của nó. Ví dụ điển hình là lớp Math.
* Với các getter/setter luôn có dạng getX, setX trong đó X là tên của thuộc tính cần được thao tác.