程序设计题 1：校园建筑用地规划

（60 分）

福建师范大学旗山校区存在多种类型的建筑用地。现要求根据分配的地理目标，采用面向对象的思想进行设计。

基类为 **Construction（建筑用地）** ，它具有所有建筑用地共有的属性和行为。其下有多个具体子类，分别代表不同类型的用地： **Ground（广场）** 、**TeachingBuilding（教学楼）** 、**AcademicBuilding（学院楼）** 和 **Canteen（食堂）** 。

每种类型的建筑用地都有其独特的功能和计算标准。你需要完成以下任务：

1. **设计类结构** ：设计一个抽象基类 Construction 和它的具体子类。
2. **实现方法** ：在基类中定义抽象方法，并在子类中提供具体实现，展示多态性。
3. **创建对象并测试** ：在 main 方法中创建不同子类的对象，统一管理并调用它们的方法。

具体要求如下：

1. **抽象类 (abstract class)** ：
   - Construction 应设计为抽象类。
   - 它应包含所有用地共有的属性（如 name, location, area, width, length）。
   - 它应包含一个实现的坐标转换方法 gcj02ToWGS84，高德坐标 GCJ02 见 excel，代码见题末。
   - 它应包含一个抽象方法 void displayInfo()，用于显示用地详细信息，由子类实现。
2. **继承 (extends)** ：
   - Ground, TeachingBuilding, AcademicBuilding, Canteen 都应继承自 Construction 类，且有自己特有的变量。
3. **构造方法 (constructor)** ：
   - 父类和子类都应提供构造方法来初始化属性。
4. **方法重写 (@Override)** ：
   - 每个子类必须重写父类的 displayInfo()方法，提供自身的具体实现。
   - 每个子类可以拥有自己特有的方法（如 Canteen 可以有一个 serveFood()方法）。
5. **多态** ：
   - 在测试类中，存储不同子类的对象，如教室按座位分为不同类型。
   - 通过循环遍历数组，调用每个对象的 displayInfo()方法，演示运行时多态。

坐标转换核心代码：

gcj02 和 wgs84 的经纬度转换方法如下：

```
private static LatLng gcj02ToWGS84(double lat, double lon) {//lat，lon 为高德纬度、经度
    // 计算偏移量
    double dLat = transformLat(lon - 105.0, lat - 35.0);//transformLat 为计算纬度偏移方法
     double dLon = transformLon(lon - 105.0, lat - 35.0);////transformLon 为计算经度偏移方法
    double radLat = lat / 180.0 * PI;
```

```
        double magic = Math.sin(radLat);
        magic = 1 - EE * magic * magic;
        double sqrtMagic = Math.sqrt(magic);
        dLat = (dLat * 180.0) / ((A * (1 - EE)) / (magic * sqrtMagic) * PI);
        dLon = (dLon * 180.0) / (A / sqrtMagic * Math.cos(radLat) * PI);
        // 计算加密后的坐标
        double mgLat = lat + dLat;
        double mgLon = lon + dLon;
        // 取原始坐标与转换后坐标的差值，得到近似 WGS-84 值
        return new LatLng(lat * 2 - mgLat, lon * 2 - mgLon);//LatLng 为封装经纬度方法
    }
```

gcj02 和 wgs84 的经纬度偏移计算方法如下：

```
    /**
     * 辅助方法：计算纬度偏移
     */
    private static double transformLat(double x, double y) {
            double ret = -100.0 + 2.0 * x + 3.0 * y + 0.2 * y * y + 0.1 * x * y + 0.2 *
Math.sqrt(Math.abs(x));
        ret += (20.0 * Math.sin(6.0 * x * PI) + 20.0 * Math.sin(2.0 * x * PI)) * 2.0 / 3.0;
        ret += (20.0 * Math.sin(y * PI) + 40.0 * Math.sin(y / 3.0 * PI)) * 2.0 / 3.0;
        ret += (160.0 * Math.sin(y / 12.0 * PI) + 320 * Math.sin(y * PI / 30.0)) * 2.0 / 3.0;
        return ret;
    }



    /**
     * 辅助方法：计算经度偏移
     */
    private static double transformLon(double x, double y) {
        double ret = 300.0 + x + 2.0 * y + 0.1 * x * x + 0.1 * x * y + 0.1 * Math.sqrt(Math.abs(x));
        ret += (20.0 * Math.sin(6.0 * x * PI) + 20.0 * Math.sin(2.0 * x * PI)) * 2.0 / 3.0;
        ret += (20.0 * Math.sin(x * PI) + 40.0 * Math.sin(x / 3.0 * PI)) * 2.0 / 3.0;
        ret += (150.0 * Math.sin(x / 12.0 * PI) + 300.0 * Math.sin(x / 30.0 * PI)) * 2.0 / 3.0;
        return ret;
    }
    /**
     * 坐标对象，封装纬度和经度
     */
    public static class LatLng {
        private final double latitude;
        private final double longitude;
        public LatLng(double latitude, double longitude) {
            this.latitude = latitude;
```

```
        this.longitude = longitude;
    }
    public double getLatitude() {
        return latitude;
    }
    public double getLongitude() {
        return longitude;
    }
    public String toString() {
        return String.format("LatLng{latitude=%.6f, longitude=%.6f}", latitude, longitude);
    }
}
```

核心参数为：

```
/** 圆周率 */
private static final double PI = Math.PI;
/** 克拉索夫斯基椭球长半轴 */
private static final double A = 6378245.0;
/** 克拉索夫斯基椭球第一偏心率平方 */
private static final double EE = 0.00669342162296594323;
/** 百度坐标系转换专用常量 */
private static final double X_PI = PI * 3000.0 / 180.0;
```

成果提交：

请用一段文字说明类的设计思路与功能规划（10 分），然后提供代码截图（40 分）和运行结果截图（10 分）。不要以文档方式。

程序设计题 2：邮件地址合法检测（40 分）

一个合法规范的邮件地址需要符合特定的格式标准，下表列出了邮件地址需遵循的核心规范和示例。

规范要素	具体要求	正确示例	错误示例
基本结构	必须包含**本地部分**、**@符号**、**域部分**，顺序固定且缺一不可	username@example.com	username.example.com(缺少@)
本地部分(@前)	可包含：字母(a-z, A-Z)、数字(0-9)、点(.)、下划线(_)、连字符(-)。 **不能**以点(.)或连字符(-)开头或结尾。 **不能**有两个或多个连续的点(..)。	user.name user_name user-name	.user@ user..name@ user-@

	长度一般不超过 64 个字符。		
@ 符号	**必须有且只能有一个**，且不能紧邻空格。	u@example.com	u@@example.com u @example.com
域部分 (@后)	需符合域名规范，通常包含点(.)分隔的多级标签。 标签可由字母、数字、连字符(-)组成， **不能**以连字符开头或结尾。 顶级域名（如.com、.cn）需有效存在。 长度不超过 253 个字符（通常）。	example.com sub.example.org	example(无点分隔) -example.com example.-com
总长度	整个邮箱地址总长通常不超过 **254 个字符** 。		
大小写	邮箱地址**不区分大小写** ，但建议**统一使用小写**以避免不必要的麻烦。（可提示或忽略）	user@example.com	USER@EXAMPLE.COM(虽等效，但不建议全部大写)

试设计一个邮件规范验证工具类 EmailValidator 来验证邮件地址是否合法，继承 Exception 类，如用户输入的邮件地址合法，输出相应提示；如用户输入的邮件地址不合法，输出具体的不合法问题提示。

请用一段文字说明类的设计思路与功能规划（10 分），然后提供代码截图（20 分）和运行结果截图（10 分）。不要以文档方式。