NJUSWI
2002-2012
Anniversary
南京大學 软件学院

防御式编程

刘 钦

南京大学软件学院

```
public class Position
{
  public Position( double latitude, double longitude )
  {
    setLatitude( latitude );
    setLongitude( longitude );
    // Default to plane geometry and kilometers
    geometry = new PlaneGeometry();
    units = new Kilometers();
  }
  public void setLatitude( double latitude )
  {
    setPhi( Math.toRadians( latitude ) );
  }
  public void setLongitude( double longitude )
  {
    setTheta( Math.toRadians( longitude ) );
  }
  public void setPhi( double phi )
  {
    // Ensure -pi/2 <= phi <= pi/2 using modulo arithmetic.
    // Code not shown.
    this.phi = phi;
  }
  public void setTheta( double theta )
  {
    // Ensure -pi < theta <= pi using modulo arithmetic.
    // Code not shown.
    this.theta = theta;
  }
  // Setters for geometry and units not shown
```

Good Design

有没有想过万一输入的时候手滑了，60写成160？

保护程序免遭非法输入数据的破坏

- 检查所有源于外部的数据的值
  - 文件、网络、用户、其他外部接口
  - 缓冲区溢出、SQL注入、注入的HTML或XML、整数溢出以及传递给系统调用的数据
- 检查子程序所有输入参数的值
- 决定如何处理错误的输入数据

断言

- 开发期间使用的，让程序在运行时进行自检的代码
- 断言为真，表面程序运行正常
- 断言为假，则意味着它已经在代码中发现了意料之外的错误。

```
public class Position
{
  public Position( double latitude, double longitude )
  {
    setLatitude( latitude );
    setLongitude( longitude );
  }
  public void setLatitude( double latitude )
  {
    // Ensure -90 <= latitude <= 90 using modulo arithmetic.
    // Code not shown.
    // Then set instance variable.
    this.latitude = latitude;
  }
  public void setLongitude( double longitude )
  {
    // Ensure -180 < longitude <= 180 using modulo arithmeti
    // Code not shown.
    // Then set instance variable.
    this.longitude = longitude;
  }
```

防御式编程