



实验5 异常处理

1 实验目的

- (1) 掌握异常处理相关概念和用法；
- (2) 具备根据需求自定义异常的能力；
- (3) 熟悉集合框架的用法，包括List、Set等。

2 实验环境

开发环境：JDK 8.0（或更高版本）+JavaFX

开发工具：Eclipse

设计工具：StarUML（或PlantUML等其他工具）

3 实验内容

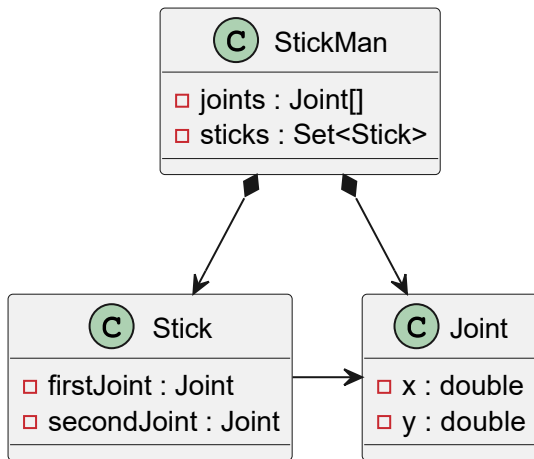
3.1 奔跑的火柴人

问题描述：设计一个身体可以移动和转动的火柴人，最终实现一个奔跑动画。



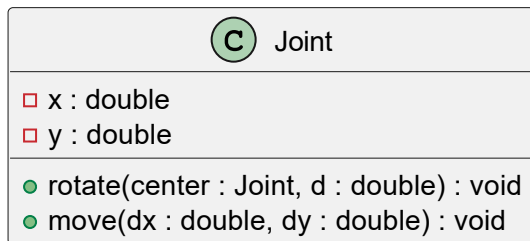
1 识别对象

火柴人（StickMan）由关节（Joint）和枝干（Stick）组成。枝干的两端是关节，在建模时可以看到一对关节。



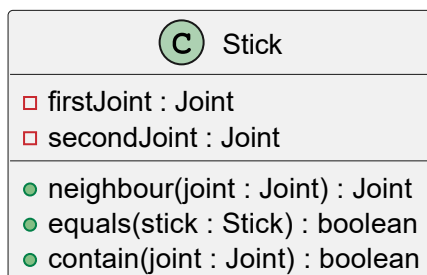
2 关节行为分析

关节主要包含转动和平移两个基本行为，转动参数包含中心点和转动弧度，平移参数包含目标在x轴和y轴移动的距离。



3 枝干行为分析

枝干连接的关节没有主次之分，一些场景需要根据给出的关节返回枝干上另外一个相邻关节。另外，枝干可能还需要支持一些判断操作，例如判断是否包含某个关节，判断两个枝干是否等价。



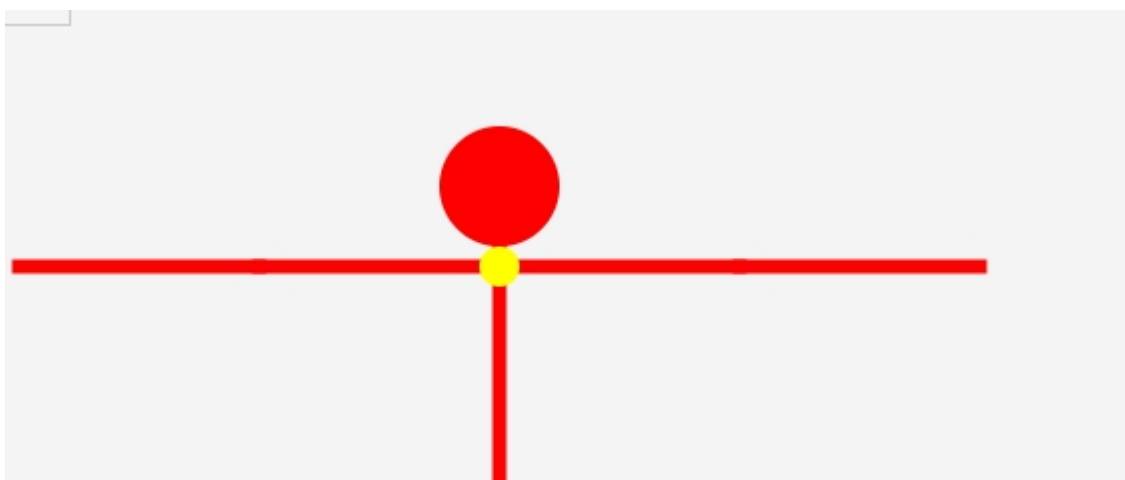
4 火柴人行为分析

火柴人的主要职责包括关节和枝干的管理以及关节的协调与控制，例如转动一个关节时，其他相关联的关节也会跟着移动。

C StickMan
<div> <div>□ joints : Joint[]</div> <div>□ sticks : Set<Stick></div> </div>
<div> <div>● rotate(center : Joint, rotated : Joint, arc : double) : void</div> <div>● move(double dx, double dy) : void</div> <div>● move(dx : double, dy : double, joints : Joint[]) : void</div> <div>● search(x : double, y : double, radius : double)</div> <div>● allConnectedJoints(target : Joint, excluded : Joint) : Set<Joint></div> <div>● existStick(j1 : Joint, j2 : Joint) : boolean</div> </div>

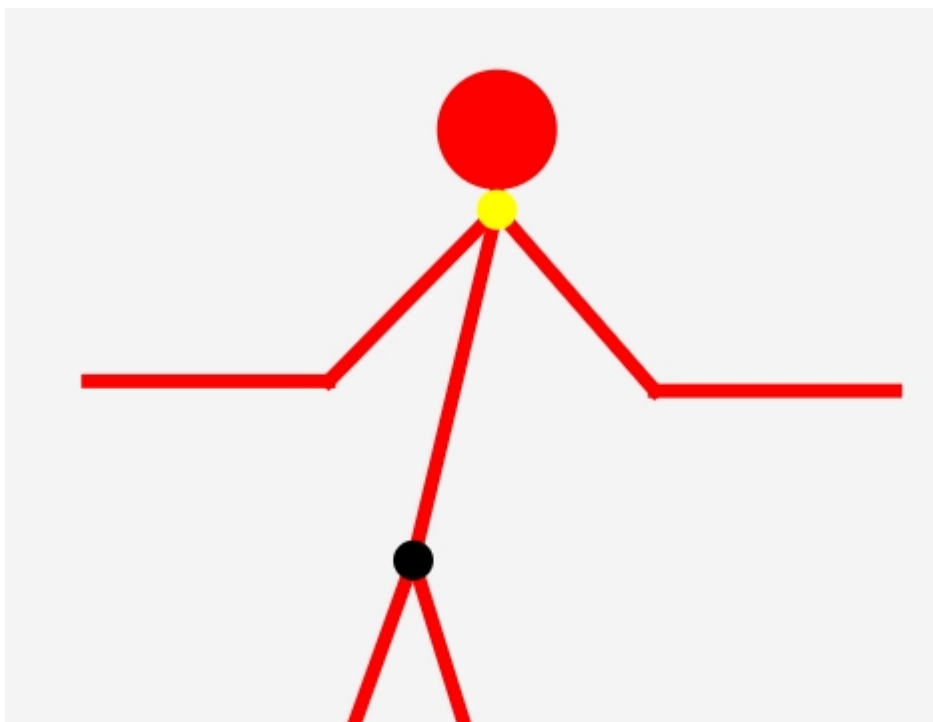
鼠标操作来选择某个关节时，坐标难以精确到浮点数，一般会提供一个容错范围或选择半径。如下图所示，根据一个圆形区域搜索最匹配的关节。

```
public Joint search(double x, double y, double radius) {}
```



当我们在操作关节运动时，其关联的关节点也会随之运动。如下图所示，在操作黄色点绕黑色点转动时，整个头部和手臂的关节均会随之移动。那么需要一个获取关联关节的方法：

```
Set<Joint> allConnectedJoints(Joint target, Joint excluded){}
```



5 火柴人控制器

这里虚构一个控制器类，主要作用是将GUI层的输入转换成火柴人的动作，并返回火柴人的状态信息。GUI层的输入主要包括鼠标移动、拖拽和单击等操作。火柴人的操作需要先选择一个关节点，即聚焦（focus）。关节点随着鼠标拖拽产生的参数进行移动（move），移动包含两种情况，一种是转动，一种是平移。转动时需要确定一个转动点（固定点），因此需要有设置固定点的操作（fix）。除了动作控制，还需要设计一些函数返回火柴人的状态，例如关节点坐标、枝干坐标等。

C Controller	
□ stickMan : StickMan;	
□ focused : Joint;	
□ fixed : Joint;	
□ tolerance : double;	
● focus(x : double, y : double) : void	
● release() : void	
● fix() : void	
● move(x1 : double, y1 : double, x2 : double, y2 : double) : void	

6 图形界面类

略：参考源代码。

7 代码实现

略：参考源代码。

8 异常处理设计

对于一个独立程序或系统一般需要构建自己的异常体系。首先将可能发生异常的情景尽可能多的列出来：

- 需要关节参与的行为，当传入的关节为空时，将无法计算，例如关节转动行为rotate(Joint center, double d)
- 若用于创建枝干的两个关节相同时，则不符合枝干的定义：Stick(Joint joint1, Joint joint2)
- 火柴人初始化后需要通过一个验证环节，例如火柴人的关节点和枝干是否完整，关节之间是否都是连通的。若没有通过基础验证，说明火柴人创建失败，应当上报异常。
- 在进行空间搜索时，搜索半径不能为负数：search(double x, double y, double radius)

在设计异常时要注意两点：

- 若能采用条件分支语句来回避异常，则尽可能不用异常来处理；
- 异常处理的语句块中不要尝试去改变业务逻辑。

以枝干创建过程为例，自定义一个枝干初始化异常。自定义异常若期望上层代码处理，则声明为checked类型，继承于Exception，否则声明为unchecked类型，继承于RuntimeException。

```
public class StickInitializationException extends RuntimeException{

    private static final long serialVersionUID = -6026447143285749889L;

    public StickInitializationException(){}
    public StickInitializationException(String msg){
        super(msg);
    }
}
```

异常的抛出：

```
public Stick(Joint joint1, Joint joint2) {  
    if(joint1 != joint2) {  
        firstJoint = joint1;  
        secondJoint = joint2;  
    }else {  
        throw new StickInitializationException("两个关节相同");  
    }  
}
```

如果需要对unchecked类型异常进行日志记录，则需要对可能发生异常的代码进行捕获和记录。

9 完善实验并整理报告

- 1、调试并运行程序；
- 2、基于UML绘制完整的类图，包括关节、枝干、火柴人、控制器四个类；
- 3、分析程序运行可能进入的异常状态，自定义异常类并设计抛出和处理代码；
- 4、在面向对象方面，程序是否存在一些改进空间，若存在则尝试简述改进方法。

4 实验要求

4.1 实验评价

- 1、必须完成实验3.1的所有步骤，实验中的程序能正常运行。
- 2、提交实验报告，作为评价的主要依据。
- 2、遇到问题时能及时与指导老师沟通并解决问题。

4.2 实验报告

本次实验需要提交实验报告至FTP服务器对应文件夹。一律采用给定的word模板编写，报告名称规范：学号+姓名，例如2023001小明。

5 实验教学录屏

https://www.bilibili.com/video/BV1jy4y1w7T1/?spm_id_from=333.999.0.0&vd_source=ab9767f8330b9092bf0b35e3238af895