Assignment Four Report

2022 2班 杨璐玮

## 实验设计：

为了体现模拟的过程以及真随机数的实现，我选择使用随机数引擎来执行模拟中一些设计为随机的参数，例如每天来参观的成年人和每个成年人携带的现金、小孩以及小孩给不同动物投喂的食物量。

首先需要确定好每一天的模拟逻辑，于是我先定义了一个模拟单天动物园运作的函数simulation\_1day()。

在simulation\_1day()函数中，根据随机数生成的成年人数量，创建成年人和他们的孩子，然后根据成年人拥有的金钱数量，为每个孩子分配购买食物的金额。接下来，根据食物的价格和剩余数量，计算每个孩子购买的食物数量，并进行喂食和扣除相应的食物库存量。然后，根据动物围栏的状态和动物的饥饿程度，决定是否喂养动物并更新相应的变量。最后，根据动物围栏的脏乱程度，决定是否关闭围栏并进行清洁。

一天的模拟实现之后，就需要进行一整个动物园周期的模拟。

整个模拟过程在simulation\_all()函数中进行simulation\_1day()的循环调用，直到动物园关闭或满足退出条件。

## 具体的类设计：

Money：记录货币金额，并实现了相关的运算符重载。

AnimalFood：记录食物的类型和数量。

AnimalEnclosure：用于容纳动物，记录脏的程度和当前状态（对游客开放/关闭）。

Animal：抽象类，包含体重和食物摄入量等属性。

Elephant：继承自Animal，包含象鼻的长度。

Giraffe：继承自Animal，包含长颈的长度。

Monkey：继承自Animal，包含手臂的长度。

Person：抽象类，包含姓名和年龄等属性。

ZooKeeper：继承自Person，包含清洁动物围栏的天数。

FoodSeller：继承自Person，包含动物食物、售出的食物的金额和静态的食物价格列表。

Visitor：继承自Person，包含游客通行证号码。

Child：继承自Visitor，包含孩子所拥有的动物食物。

Adult：继承自Visitor，包含所拥有的金钱。

## 遇到的问题/难点：

* 对象的数据成员会从初始化的量开始重新计算，反复重置

为了解决这个问题，必须把主循环函数的值传递改为引用传递，避免重复调用拷贝构造函数从而反复初始化参数。

* 清理的时候，如何设定和判断动物围栏隔天开放？

我设计使用了一个隔天比较的算法。关闭围栏的时候，记录下闭关的天数，把围栏的状态设为关闭。第二天如果检查到围栏是关闭，但是第二天的天数的奇偶与记录下来的天数不符合，于是保持关闭状态；第三天的时候奇偶相同了，于是重新开放。

* 参数问题

尽管程序实现了随机，但是实际上不是理想状态下的随机。闭园的原因很大程度上取决于固定参数的设置，比如食物总数和zookeeper清理围栏的dirtylevel阈值。两者相对来说谁更高，谁的可能性更大一些。而在给定的参数下，总会有对应的几种情况反复出现。

* Zookeeper工作时长的问题

在文档中规定了清理围栏需要关闭一天的时间，这个时候由于模拟的随机规则和dirtylevel的设置，大概率会有多个围栏同时出现需要清理的情况。这个时候如果按照设定，zookeeper需要在一天内完成多个清理工作，但是都会被计入工作时长，导致实际开园8天而工作10天的神奇现象。解决方案有两个：

调整固定的参数，人为限制大概率不会出现多个围栏同时需要清理的时候；

提高zookeeper的工作效率，即使出现多个围栏需要清理的时候，让他都在一天内完成工作。只需要把每次需要清理的时候直接增加工作时长改为改变某个标志变量flag，如果flag为真，在每一天模拟结束后，给zookeeper增加工时一天。

## 部分模拟效果展示：









