

# ThoughtWorks®

OO training

---

## PARKING LOT

---

郑培真 ([pzzheng@thoughtworks.com](mailto:pzzheng@thoughtworks.com))  
周颖([ybzhou@thoughtworks.com](mailto:ybzhou@thoughtworks.com))

# WHAT WE FOCUS ON

---

- ▶ **OO Principles**
- ▶ **TDD**
- ▶ **Tasking**
- ▶ **Refactoring**
- ▶ **Design Pattern**

# WHAT WE FOCUS ON

---

## ► OO Principles

1. encapsulation: 控制 accessor, mutator 的访问权限
2. abstraction: 对真实事务的抽象，表现为类的识别，**接口**的定义。
3. inheritance: is-a 关系
4. polymorphism: 多态，“one name, many forms”，手段有重写（继承/实现）、重载。  
一种 high level 的对象管理
5. S.O.L.I.D: 易维护、易扩展
6. favor composition over inheritance...

## ► TDD

## ► Tasking

## ► Refactoring

## ► Design Pattern

# WHAT WE FOCUS ON

---

- ▶ **OO Principles**

- ▶ **TDD**

1. red-green-refactor
2. 小步
3. 仅当测试失败再写代码
4. 多从业务角度读测试代码

- ▶ **Tasking**

- ▶ **Refactoring**

- ▶ **Design Pattern**

# WHAT WE FOCUS ON

---

- ▶ **OO Principles**

- ▶ **TDD**

- ▶ **Tasking**

1. 先有 task, 再开始开发
2. 从业务角度列 task, 增量迭代式
3. 可测
4. 控制在有限时间内

- ▶ **Refactoring**

- ▶ **Design Pattern**

# WHAT WE FOCUS ON

---

- ▶ **OO Principles**

- ▶ **TDD**

- ▶ **Tasking**

- ▶ **Refactoring**

1. 三次原则：如果相同代码写第二遍，根据情况重构；写第三遍，必须重构
2. 重构包括测试和功能代码
3. 什么时候重构？
  1. 实现功能后，识别 bad smell，重构
  2. 接到新需求，新需求实现复杂，先重构
4. 十六字心法：旧的不变，新的创建，一步切换，旧的再见

- ▶ **Design Pattern**

# WHAT WE FOCUS ON

---

- ▶ **OO Principles**

- ▶ **TDD**

- ▶ **Tasking**

- ▶ **Refactoring**

- ▶ **Design Pattern**

1. 工厂模式、策略模式、命令模式、观察者模式...
2. 能用尽量用
3. 如果不熟悉，也可以通过不断重构，重构到设计模式

# 停车场模拟系统



# PARKING LOT

---

停车场可以停车  
停车场可以取车

# PARKING BOY

---

作为一个停车小弟（ParkingBoy），我能够将车顺序停放到多个停车场，并可以取出

# SMART PARKING BOY

---

作为一个聪明的停车小弟（Smart Parking Boy），我能够将车停在**空车位最多**的那个停车场

# SUPER PARKING BOY

---

作为一个超级停车小弟（Super Parking Boy），我能够将车停在**空置率**最高的那个停车场

# ASSIGNMENTS

---

作为停车场的经理（Parking Manager），我要管理多个停车仔，让他们停车，同时也可以自己停车；

选择停车仔/停车场时，按顺序选择；

可以管理各种各样的停车仔