



尚马教育

## 项目案例：迷你共享单车



- 能理解程序基本概念 - 程序、变量、数据类型
- 会使用顺序、选择、循环、跳转语句编写程序
- 会使用数组、操作字符串
- 会使用带参方法
- 会定义类、创建和使用对象
- 使用SimpleDateFormat类对字符串进行日期格式化

## ❖ 系统概述

➤ 为某企业开发一个迷你**共享单车**管理器，实现单车的管理，包括如下功能：

- ◆ 投放Bike
- ◆ 查看Bike
- ◆ 删除Bike
- ◆ 借出Bike
- ◆ 归还Bike

```
Console
App [Java Application] C:\Program Files\Java\jre1.8.0_162\bin\javaw.exe (2019年8月12日 下午4:18:
欢迎使用迷你共享单车管理系统
*****
1, 投放Bike
2, 查看Bike
3, 删除Bike
4, 借出Bike
5, 归还Bike
6, Bike排行榜
7, 退    出
*****
请选择:2
----->查看所有的Bike信息
```

序号	品牌	数量	借出次数	
1	ofo单车	3	100	
单车编号	单车名称	单车状态	借出时间	
1	ofo单车0	可借		
2	ofo单车1	可借		
3	ofo单车2	可借		

  

序号	品牌	数量	借出次数	
2	halo单车	3	210	
单车编号	单车名称	单车状态	借出时间	
1	halo单车0	可借		
2	halo单车1	可借		
3	halo单车2	可借		



- 系统开发步骤
  - 明确需求
  - 设计对象保存Bike信息
  - 设计技术框架
    - 采用Java技术
  - 编码顺序
    - 1、数据初始化
    - 2、菜单切换
    - 3、完成各种bike管理操作
  - 测试



- 界面交互设计的原则
  - 统一性原则
    - 界面风格统一：  
用相同方式展现相同类型的数据，如：日期类型
    - 交互风格统一：  
用相同方式完成相同类型的操作，如：录入日期
  - 美观性原则
    - 界面美观大方
  - 易用性原则
    - 操作方式自然、易理解

- 如何保存Bike信息？
- 如何访问Bike信息？
- 如何删除Bike信息？
- 如何计算租金？



- 使用数组对象保存Bike信息
  - 每个Bike的信息都包括名称、是否可借的状态以及借出的日期
  - 包含多个Bike
- 参考代码

```
public class SharedBikeCompany {  
    private Integer bikeCompanyId;// 公司id  
    private String bikeCompanyName;// ofo 哈罗 摩拜  
    private SharedBike[] sharedBikes;// 公司持有共享单车  
    private Integer sum;//公司单车总量  
    private Integer count;// 公司单车借出次数  
}
```

```
public class SharedBike {  
    private Integer bid;// 单车编号  
    private String bname;// 单车名称-->ofo 哈罗 摩拜  
    private Integer status;// 单车状态--> 0 已借出 1 可借状态  
    private String borrowTime;// 单车借出时间  
}
```



- 访问Bike信息
  - 创建company对象后，访问第i+1个bike信息
    - 名称： `company.getSharedBikes()[i].getBname()`
    - 状态： `company.getSharedBikes()[i].getStatus()`
- 参考代码

```
SharedBikeCompany[] bikeCompanies = new SharedBikeCompany[3];  
...  
SharedBike bike = company.getSharedBikes()[i];  
String status = (bike.getStatus() == 1) ? "可借" : "已借出";  
String borrowTime = (bike.getBorrowTime() == null) ? " " : bike.getBorrowTime();  
...
```





- 计算租金
  - 租金=日期差\*（租金/每天）
  - 计算日期差
    - 借出日期利用“年-月-日”的字符串形式保存，可利用类 SimpleDateFormat 对字符串进行日期格式化，再求日期差
- 参考代码

```
SimpleDateFormat sd=new SimpleDateFormat("yyyy-MM-dd");  
...  
Date d1=sd.parse(dstr1);  
Date d2=sd.parse(dstr2);  
charge=(d2.getTime()-d1.getTime())/(24*60*60*1000);//得到日期差
```



- 用例1：数据初始化[20分钟]
- 用例2：实现菜单切换[25分钟]
- 用例3：实现查看Bike信息[20分钟]
- 用例4：实现新增Bike信息[20分钟]
- 用例5：实现删除Bike信息[25分钟]
- 用例6：实现借出Bike业务处理[25分钟]
- 用例7：实现归还Bike业务处理[25分钟]
- 用例8：实现借出排行榜[20分钟]

bname	status	borrowtime
of0单车1	0	2019-8-12 12:00:00
halo单车2	1	
mobai单车0	1	

- 需求说明
  - 初始化Bike信息
- 思路分析
  - 创建项目bike
  - 创建类SharedBike/SharedBikeCompany，添加相应属性
  - 创建类SharedBikeMgr，添加方法initial()，初始化3个公司，以及每个公司单车基本信息。



- 功能测试
  - 创建类SharedBike/SharedBikeCompany
  - 创建类SharedBikeMgr，初始化单车信息

功能测试



- 需求说明

- 编写程序入口，实现菜单显示和切换

- 思路分析

- 创建方法startMenu(), 实现菜单切换
- 创建方法returnMain(), 返回主菜单
- 编写类Start, 实现程序入口

- 难点提示

- 使用switch语句实现菜单切换

App [Java Application] C:\Program Files\Java\jre1.8.0\_162\bin\javaw.exe

欢迎使用迷你共享单车管理系统

\*\*\*\*\*

- 1, 投放Bike
- 2, 查看Bike
- 3, 删除Bike
- 4, 借出Bike
- 5, 归还Bike
- 6, Bike排行榜
- 7, 退 出

\*\*\*\*\*

请选择:1

----->1. 投放Bike

-----

输入0返回:0

\*\*\*\*\*

- 1, 投放Bike
- 2, 查看Bike
- 3, 删除Bike
- 4, 借出Bike
- 5, 归还Bike
- 6, Bike排行榜
- 7, 退 出

\*\*\*\*\*

请选择:2



- 功能测试
  - 实现菜单切换
  - 输入非数字时程序报错

功能测试



- 需求说明
  - 遍历所有公司下的Bike，获取数组元素信息，并进行显示
- 思路分析
  - 在类SharedBikeDao中创建方法selectAllBike()，完成功能要求
  - 根据状态值输出“已借出”或“可借”
- 难点提示
  - 循环结束条件：
  - `bikeCompanies[i].getSharedBikes()[i]==null`
  - 避免变量为null

- 功能测试

- 正确输出所有Bike信息
- 正确输出Bike状态信息
- 输出格式正确

App [Java Application] C:\Program Files\Java\jre1.8.0\_162\bin\javaw.exe (2019年8月12日 下午5:27:49)

公司序号	公司名称	公司单车数量	公司单车借出次数
1	ofo单车	3	100

单车序号	单车品牌	单车状态	借出时间
1	ofo单车0	可借	
2	ofo单车1	可借	
3	ofo单车2	可借	

公司序号	公司名称	公司单车数量	公司单车借出次数
2	halo单车	3	210

单车序号	单车品牌	单车状态	借出时间
1	halo单车0	可借	
2	halo单车1	可借	
3	halo单车2	可借	

公司序号	公司名称	公司单车数量	公司单车借出次数
3	摩拜单车	3	50

功能测试





- 需求说明
  - 向bike数组新增n辆某品牌的单车信息，实现新增Bike信息

- 思路分析
  - 在类SharedBikeDao中创建方法putBike()，完成功能要求

- 定位新增位

- 插入Bike

- 难点提示

- 定位新增位

```
*****
请选择:1
----->1.投放Bike

1:ofo单车
2:halo单车
3:摩拜单车
请选择要投放的单车品牌
2
请录入要投放的数量:
5
空间不足，已经参照配置进行扩容
投放5辆halo单车成功
-----
输入0返回:0
*****
```



- 功能测试

- 正确新增Bike

公司序号	公司名称	公司单车数量	公司单车借出次数
2	halo单车	3	210

单车序号	单车品牌	单车状态	借出时间
1	halo单车0	可借	
2	halo单车1	可借	
3	halo单车2	可借	

公司序号	公司名称	公司单车数量	公司单车借出次数
2	halo单车	8	210

单车序号	单车品牌	单车状态	借出时间
1	halo单车0	可借	
2	halo单车1	可借	
3	halo单车2	可借	
4	halo单车3	可借	
5	halo单车4	可借	
6	halo单车5	可借	
7	halo单车6	可借	
8	halo单车7	可借	

- 需求说明

- 按照输入的名称/编号，删除指定Bike

- 思路分析

- 查找要删除元素
  - 通过把后面的元素依次前移一
  - 最后一个不为空元素置空

- 难点提示

- 不允许删除借出状态的Bike
  - 不存在指定Bike，给出提示

```
=====
请选择:3
----->3.删除Bike

1:ofo单车
2:halo单车
3:摩拜单车
请选择要操作的单车品牌
2
此halo单车有共享单车如下:
单车序号 |      单车品牌      | 单车状态 | 借出时间
1         |      halo单车0     | 可借     |
2         |      halo单车1     | 可借     |
3         |      halo单车2     | 可借     |

请录入要删除单车的编号:
1
删除halo单车公司下的halo单车0成功
-----
输入0返回:0
=====
```



### ● 功能测试

- 正确删除未借出Bike
- 删除借出状态的Bike，给出
- 指定删除不存在的Bike，给出相应提示

公司序号	公司名称	公司单车数量	公司单车借出次数
2	halo单车	3	210

  

单车序号	单车品牌	单车状态	借出时间
1	halo单车0	可借	
2	halo单车1	可借	
3	halo单车2	可借	

公司序号	公司名称	公司单车数量	公司单车借出次数
2	halo单车	2	210

  

单车序号	单车品牌	单车状态	借出时间
1	halo单车1	可借	
2	halo单车2	可借	

功能测试

- 需求说明
  - 按照输入的编号，借出指定Bike

- 思路分析
  - 查找要借出的Bike
  - 判断是否存在
  - 判断是否已借出
  - 如果可借，修改状态值

- 难点提示
  - 各种情况的判断条件

```
请选择:4  
----->4. 借出Bike
```

```
1: ofo单车  
2: halo单车  
3: 摩拜单车  
请选择要操作的单车品牌
```

```
2  
此halo单车有共享单车如下:  
单车序号 | 单车品牌 | 单车状态 | 借出时间  
1 | halo单车0 | 可借 |  
2 | halo单车1 | 可借 |  
3 | halo单车2 | 可借 |
```

```
请录入要借出的单车编号:1  
请输入借出日期(2019-01-01 12:30:30)2019-08-02 12:30:30  
借出halo单车公司下的<<halo单车0>>成功
```

```
-----  
输入0返回:0  
.....
```



### ● 功能测试

- 正确借出匹配的才
- 借出匹配的已借出
- 借出无匹配的Bik

公司序号	公司名称	公司单车数量	公司单车借出次数
2	halo单车	3	210
单车序号	单车品牌	单车状态	借出时间
1	halo单车0	可借	
2	halo单车1	可借	
3	halo单车2	可借	

公司序号	公司名称	公司单车数量	公司单车借出次数
2	halo单车	3	210
单车序号	单车品牌	单车状态	借出时间
1	halo单车0	已借出	2019-08-013 12:30:30
2	halo单车1	可借	
3	halo单车2	可借	



- 需求说明

- 编写方法 `returnBike()`，实现

公司序号	公司名称	公司单车数量	公司单车借出次数
2	halo单车	3	210

  

单车序号	单车品牌	单车状态	借出时间
1	halo单车0	可借	
2	halo单车1	已借出	2019-01-01 12:30:30
3	halo单车2	可借	

- 思路分析

- 查找要归还的Bike
  - 判断是否存在
  - 判断是否未借出
  - 如果已借出，修改sta

- 难点提示

- 字符串-日期转换

```
请选择:5
----->5.归还Bike
```

```
1:of0单车
2:halo单车
3:摩拜单车
请选择要操作的单车品牌
```

```
2
请录入要归还的单车编号:2
请录入归还时间(yyyy-MM-dd HH:mm:ss):2019-01-01 14:30:30
您的借车时间为:2019-01-01 12:30:30
用车时间为<<2>>小时，需要支付:2元
```

```
-----
输入0返回:0
```



- 功能测试

- 正确归还借出状态的Bike
- 归还未借出的Bike，给出相应提示
- 归还不存在的Bike，给出相应提示

公司序号	公司名称	公司单车数量	公司单车借出次数
2	halo单车	3	211

  

单车序号	单车品牌	单车状态	借出时间
1	halo单车0	可借	
2	halo单车1	可借	
3	halo单车2	可借	





- 需求说明
  - 增加菜单项“借出排行榜”，实现借车次数从大到小的顺序显示借出排行榜
- 思路分析
  - 利用冒泡排序实现排行榜
  - 每次借出成功后，count++
- 难点提示
  - 冒泡算法

```
***** /, 赵 山 *****
请选择:6
|----->6. 排行榜
公司名称      借车次数
halo单车      210
ofo单车       100
摩拜单车      50
-----
输入返回
```



- 冒泡排序算法

- 依次比较相邻两个数，将小数放前，大数放后。需比较 $n-1$ 趟
- 第一趟：依次比较第1个数和第2个数、第2个数和3个数，直至最后两个数。第一趟结束，将最大数放到了最后
- 第二趟：依次比较第2个数和第3个数、第3个数和4个数，直至最后两个数。
- 依次进行第三趟...

- 冒泡排序示例

```
public static void main(String[] args) {  
    int arr[] = { 10, 20, 50, 40, 23, 45, 60 };  
    for (int i = 0; i < arr.length-1; i++) {  
        for (int j = i + 1; j < arr.length; j++) {  
            if (arr[i] > arr[j]) {  
                //交换  
                int temp = arr[i];  
                arr[i] = arr[j];  
                arr[j] = temp;  
            }  
        }  
    }  
    //输出排序后数组  
    for (int i = 0; i < arr.length; i++) {  
        System.out.println(arr[i]);  
    }  
}
```

利用冒泡排序算法进行排序



- 功能测试
  - 正确输出借车排行榜
  - 输出格式正确

功能测试

## 项目展示

- 讲解要点：

- 完成情况、技能总结、经验分享、项目收获

- 表达要求：

- 清晰流畅、有条理、重点突出



- 主要技能点
  - 选择结构、循环结构、二重循环结构
  - 数组
  - 类和对象
  - SimpleDateFormat 类
- 项目流程
  - 需求→设计→开发→测试→部署→用户培训
  - 设计：数据库设计、技术框架设计、交互设计