**每日作业卷**

JavaSE--day04【内部类、static、包、访问修饰符、final】

# 关卡1

## 训练案例1

### 训练考核知识点

static关键字

### 训练描述

请阐述对static关键字理解

### 操作步骤描述

1. 请说出 static 可以修饰什么?
2. 请说出 static 关键字的特点 ?
3. 请说出 static 修饰的成员如何使用 ?

## 训练案例2

### 训练考核知识点

静态代码块

### 训练描述

请阐述你对静态代码块的理解

### 操作步骤描述

1. 请说出静态代码块的定义位置.
2. 请说出静态代码块的定义格式.
3. 请说出静态代码块执行时机和执行次数.
4. 请说出静态代码块可用于做什么?

## 训练案例3

### 训练考核知识点

final关键字

### 训练描述

请阐述你对final关键字的理解

### 操作步骤描述

1. 请说出final关键字可以修饰什么?

2. 请说出被final关键字的特点.

## 训练案例4

### 训练考核知识点

包

### 训练描述

请阐述你对包的理解

### 操作步骤描述

1. 请说出包的作用
2. 请说出包命名规则
3. 请说出包声明格式与位置
4. 请说出如何访问包中的类
5. 请说出导包的作用和导包的格式以及导包语句的位置

## 训练案例5

### 训练考核知识点

四种访问权限修饰符

### 训练描述

请阐述你对四种访问权限修饰符的理解

### 操作步骤描述

1. 请说出四种权限修饰符分别是什么 ?
2. 请说出四种权限修饰符可以修饰什么?
3. 请说出每种权限修饰符特点 ?
4. 请说出如何使用每种权限修饰符?

## 训练案例6

### 训练考核知识点

内部类

### 训练描述

请阐述内部类的概念

### 操作步骤描述

1. 请说出什么是内部类.
2. 请说出什么时候使用内部类.
3. 请说出内部类的分类.

## 训练案例7

### 训练考核知识点

成员内部类

### 训练描述

请阐述你对成员内部类的理解

### 操作步骤描述

1. 请说出成员内部类定义位置.
2. 请说出成员内部类定义格式.
3. 请说出如何通过成员内部类创建对象.

## 训练案例8

### 训练考核知识点

匿名内部类

### 训练描述

请阐述你对匿名内部类的理解

### 操作步骤描述

1. 请说出匿名内部类的两个含义分别是什么?.
2. 请说出匿名内部类的作用.

## 训练案例9

### 训练考核知识点

引用数据类型

### 训练描述

请阐述你引用数据类型作为参数和返回值类型理解

### 操作步骤描述

1. 请说出你对类作为参数和返回值类型理解.
2. 请说出你对抽象类作为参数和返回值类型理解
3. 请说出你对接口作为参数和返回值类型理解

## 训练案例10

### 训练考核知识点

静态成员变量

### 训练描述

请说出下面代码执行结果并绘制内存图

**public** **class** Student {

// 学校

**public** **static** String *school* = "清华大学";

// 姓名

**private** String name;

// 年龄

**private** **int** age;

// 构造方法

**public** Student() {

**super**();

}

**public** Student(String name, **int** age) {

**super**();

**this**.name = name;

**this**.age = age;

}

}

**public** **class** Test {

**public** **static** **void** main(String[] args) {

Student s1 = **new** Student("景甜",22);

Student s2 = **new** Student("林思意",18);

System.***out***.println(s1.*school*);

s1.*school* = " 极客营";

System.***out***.println(s2.*school*);

}

}

### 操作步骤描述

1. 请说出上面代码的运行结果
2. 请画出上面的代码的内存图
   1. 打开画图软件
   2. 在左上角写上:静态成员变量的内存图
   3. 在左边画一个竖长的文本框,把字体调整到10号,把上面的代码拷贝过去
   4. 在代码的右边绘制一个方块表示栈区,右边需要留出绘制堆内存区域,下面需要留出绘制方法区的区域
   5. 在栈内存的方法,绘制一个表示堆内存的方块,在该方块左上角表上堆
   6. 在栈内存和堆内存方块下面绘制一个方块表示方法区,在左上角标上方法区
   7. 在方法区左边绘制一个方块,表示为Test.class代码分配的内存,方法左上角标注Test.class; 方法内部写Test.class的部分代码.
   8. 在栈区最下面画一个方块,左上角标注main方法,表示main方法压栈执行
   9. 在方法区Test.class区域,右边绘制一个方块,左上角标注为Student.class 表示为Student.class代码分配的内存, 方块内部可以写Student类的部分代码
   10. 在方法区Student.class方法的右边绘制一个方块,左上角标注静态区
   11. 在静态区,画一个方块,左上角写上 Student.class 表示Student.class中的静态变量内存
   12. 在Student.class方块内,画一个方块,左上角标注 static school,表示为静态成员school分配的存储空间, 在这个方块内部写 “清华大学” ,表示默认初始化值为清华大学
   13. 在main方块中,上面绘制一个小方块,左上标注为Student s1 表示为s1分配的存储
   14. 在堆内存中,绘制一个方块,右上角标注new Student(“景甜”,22),表示创建对象分配的存储空间,在这个小方块内绘制绘制两个小方块一个左上标注为name,表示为成员变量name分配的存储空间,内部写上”景甜”,一个标注为age,内部写上22,表示为age成员变量分配的存储空间,初始化值为22
   15. 在这个对象的方块,左上角标注上 0X001,表示这个对象的地址;在栈内存中,方块s1中写上0X001,表示变量中存储是对象的地址;画一条从这里到堆内存的0X001的线,然后标注上箭头(两根线),表示s1指向堆内存.
   16. 在main方块中,上面绘制一个小方块,左上标注为Student s2 表示为s2分配的存储
   17. 在堆内存中,绘制一个方块,右上角标注new Student(“林思意”,18),表示创建对象分配的存储空间,在这个小方块内绘制绘制两个小方块一个左上标注为name,表示为成员变量name分配的存储空间,内部写上”林思意”,一个标注为age,内部写上18,表示为age成员变量分配的存储空间,初始化值为18
   18. 在这个对象的方块,左上角标注上 0X002,表示这个对象的地址;在栈内存中,方块s2中写上x003,表示变量中存储是对象的地址
   19. 使用红框把System.***out***.println(s1.*school*);中的s1.school 框起来,画一条从这里到线到静态区school的线条,标注上箭头(使用两根线),表示s1.school访问的是静态区school,所以此处输出的是: 清华大学
   20. 使用红框把s1.*school* = " 极客营"; 中的s1.school框起来,然后一个从这里到静态区school的箭头,然后划掉school静态变量的清华大学,在后面写上 极客营,表示把静态变量school的从原来的清华大学修改为极客营
   21. 使用红框把System.***out***.println(s2.*school*);中的s2.school 框起来,画一条从这里到线到静态区school的线条,标注上箭头(使用两根线),表示s2.school访问的是静态区school,所以此处输出的是: 极客营

## 训练案例11

### 训练考核知识点

静态代码块

### 训练描述

看下面的代码说出运行结果以及执行流程

**public** **class** Test {

**static** {

System.***out***.println("Test 静态代码块");

}

**public** Test(){

System.***out***.println("Test 构造方法");

}

**public** **static** **void** main(String[] args) {

System.***out***.println("Test main方法");

Test t = **new** Test();

}

}

### 操作步骤描述

1. 请说上面代码运行结果
2. 请说出上面代码的执行流程

## 训练案例12

### 训练考核知识点

成员内部类

### 训练描述

请用代码描述:

汽车有两种状态,一种是停止,一种运行; 汽车的内部都一个发动机;发动机有一个工作的功能; 如果汽车的运行状态发动机就飞速旋转,如果汽车是停止状态,发动机停止工作

要求: 使用成员内部类.

### 操作步骤描述

1. 定义一个汽车类(Car)
   1. 成员变量: 状态(status),true表示运行状态一种,false表示停止状态.
   2. 成员内部类: 发动机(Engine)
      1. 成员方法:work()
         1. 如果汽车的运行状态打印发动机就飞速旋转
         2. 否则打印发动机停止工作
   3. 提供:状态(status)的getter和setter方法
2. 定义测试类(Test)
   1. 提供main方法
   2. 在main方法中
      1. 创建汽车对象 c
      2. 通过汽车对象c创建该汽车的发动机对象e
      3. 调用发动机e对象的work()方法
      4. 把汽车对象c的状态设置为true
      5. 调用发动机e对象的work()方法

# 关卡2

## 训练案例1

### 训练考核知识点

权限修饰符

static

final

abstract

### 训练描述

请简述abstract不能和那些关键一起使用,为什么.

### 操作步骤描述

1. abstract可以与private关键字一起使用吗? 为什么?
2. abstract 可以与static关键字一起使用吗? 为什么?
3. abstract 可以与final关键字一起使用吗? 为什么?

## 训练案例2

### 训练考核知识点

静态代码块

### 训练描述

看下面的程序,说出的它的执行结果,以及执行流程

**public** **class** Student {

**static** {

System.***out***.println("Student 类的静态代码块");

}

**public** Student(){

System.***out***.println("Student 类的构造方法");

}

**public** Student(String methodName){

System.***out***.println("在"+methodName+"方法中创建Student对象调用的");

}

}

**public** **class** Test {

**static** {

System.***out***.println("Test 类的静态代码块");

}

**public** **static** **void** main(String[] args) {

System.***out***.println("执行main方法");

// 创建学生对象

Student s1 = **new** Student("main");

}

}

提示: 类只有被使用到的时候才会加载

### 操作步骤描述

1. 说出输出结果
2. 阐述类的加载与代码的执行流程

## 训练案例3

### 训练考核知识点

final关键字

### 训练描述

请写代码验证final关键字的特点

### 操作步骤描述

1. 验证:final修饰类不可以被继承，但是可以继承其他类
   1. 定义类(Final\_1),用于验证第一个特点
   2. 在类上面使用注释写上: 验证:final修饰类不可以被继承，但是可以继承其他类
   3. 在Final\_1.java中,定义被修final修饰的类(FinalClass) 类
   4. 在Final\_1.java中,定义子类Sub继承FinalClass类,编译报错,把鼠标放在红色X上面,提示:The type Sub cannot subclass the final class FinalClass:
   5. 把类Sub,使用注释掉,然后在下面写上: 被final修饰的类不能被继承
   6. 让FinalClass继承Final\_1,编译通过; 在FinalClass上面使用注释写上: 验证:被final的修饰类可以继承其他类.
2. final修饰的方法不可以被覆盖,但父类中没有被final修饰方法，子类覆盖后可以加final
   1. 定义类(Final\_2),用于验证第二特点
      1. 成员方法:
         1. 被final修饰的成员方法 finalMethod
         2. 普通方法 normalMethod
   2. 在Final\_2.java中,定义子类(Sub2) 继承 Final\_2
      1. 覆盖 finalMethod编译报错,把鼠标放到红色x上面,提示:Cannot override the final method from Final\_2: 被final修饰方法不能被覆盖(重写);
      2. 注释调用这个方法,在上面使用注释写上: final修饰方法不能被覆盖(重写);
      3. 覆盖normalMethod 方法,在权限修饰符 和返回值直接加上final,编译通过,在这个方法上面写上: 父类中没有被final修饰方法，子类覆盖后可以加final
3. final修饰的变量称为常量，这些变量只能赋值一次
   1. 定义Final\_3类,表示用于验证final的第三个特点
   2. 在这个类上面使用注释写上: 验证: final修饰的变量称为常量，这些变量只能赋值一次
   3. 提供main方法
   4. 在main方法中
      1. 定义被final修饰的整型变量a,初始化值为10
      2. 给a赋值为20,编译报错,
      3. 把鼠标放在红色x上面提示:The final local variable a cannot be assigned:被fianl修饰的局部变量不能被赋值;
      4. 注释掉这句话,在上面写上:final修饰的变量称为常量，这些变量只能赋值一次
      5. 定义被final修饰的整型变量b,没有初始化值
      6. 下一行,给变量b赋值为5
      7. 然后再给变量b赋值为15,编译报错
      8. 把鼠标放到红色x上面提示:The final local variable b may already have been assigned: 被final修饰的局部变量已经赋值了
      9. 注释掉这个赋值语句,然后在上面使用注释写上:final修饰的变量称为常量，这些变量只能赋值一次
4. 引用类型的变量值为对象地址值，地址值不能更改，但是地址内的对象属性值可以修改
   1. 定义类(Final\_4),表示用于验证final的第四个特点,在这个类上面使用注释写上:验证:引用类型的变量值为对象地址值，地址值不能更改，但是地址内的对象属性值可以修改
   2. 在Fianl4.java中,定义辅助类,随便写,我这里使用猫类(Cat)
      1. 定义公共权限成员变量: 年龄 (age)
   3. 在Fianl\_4类中提供main方法
   4. 在main方法中
      1. 定义被final修饰的Cat类型的引用变量c,new Cat()赋值给它
      2. 然后,在new Cat() 赋值给变量c,编译报错
      3. 错误提示:The final local variable c cannot be assigned. 被fianl修饰的局部变量不能被赋值
      4. 注释掉这个赋值语句,然后在上面使用注释写上: 引用类型的变量值为对象地址值，地址值不能更改
   5. 给对象c的age属性赋值为2,编译通过; 表示: 其指向的对象属性值可以修改
   6. 在这话上面,使用注释写上:地址内的对象属性值可以修改
5. 修饰成员变量，需要在创建对象前赋值，否则报错。(当没有显式赋值时，多个构造方法的均需要为其赋值。
   1. 定义类(Final\_5),表示用于验证final的第五个特点
   2. 在这个类上面使用注释写上:验证:修饰成员变量，需要在创建对象前赋值，否则报错。(当没有显式赋值时，多个构造方法的均需要为其赋值。
   3. 这里需要验证两点:
      1. 定义final成员变量的时候,显示赋值
      2. 否则每一个构造方法都需要给这个final修饰成员变量赋值
   4. 定义被final修饰的整型成员变量a,并显式赋值为10
   5. 定义被final修饰的整型成员变量b,不显式赋值,编译报错,
      1. 提示The blank final field b may not have been initialized,被final修饰的成员变量可能没有被赋值
   6. 提供空参构造方法,上面成员变量b不报错了,但是这里开始报错了
      1. 提示:The blank final field b may not have been initialized
      2. 在空参构造方法内部写上: b = 20; 编译通过
   7. 提供满参构造,但是不给a和b赋值,报错
      1. 提示:The blank final field b may not have been initialized
      2. 在满参构造方法中写上this.b = b; 编译通过,说明: 当没有显式赋值时，多个构造方法的均需要为其赋值.
      3. 在写上 this.a = a; 编译报错,提示:The final field Final\_5.a cannot be assigned 表示被final修饰成员变量只能赋值一次
      4. 注释掉这句代码,在后面写上,被final修饰成员变量只能赋值一次

## 训练案例4

### 训练考核知识点

抽象类

匿名内部类

### 训练描述

请用代码描述:

在某新型公司,每一个员工都有工号,姓名; 都要工作,但是每一个员工的工作内容都不一样;

要求:使用匿名内部类创建对象

### 操作步骤描述

1. 定义抽象类(Employee)
   1. 属性: 工号(id),姓名(name)
   2. 行为: 抽象方法work()
   3. 要求: 提供setters和gettters方法
2. 定义测试类Test
   1. 提供main方法
   2. 在main方法中
      1. 使用匿名内部类创建Employee的子类对象 e;并重写work()方法
         1. 输出格式:工号为001的员工范冰冰在努力的敲代码
      2. 使用setXxx方法给e的工号和名称赋值
      3. 调用e的work()方法

## 训练案例5

### 训练考核知识点

接口

接口名作为形式参数类型

匿名内部类

### 训练描述

请用代码描述：

某市举办一场运动会,只要是会跑的都可以参加.

要求: 使用到接口,接口作为参数,匿名内部类.

提示: 把跑功能抽取到接口中 ,在测试类中提供参加运动会的静态方法

### 操作步骤描述

1. 定义接口(Sport)
   1. 抽象方法: 跑(run)
2. 定义测试类Test
   1. 定义静态方法:进入运行会(enter(Sport s)),在方法中调用s的run方法
   2. 提供main方法
   3. 在main方法中
      1. 调用enter(Sport s),传入使用Sport接口的匿名内部类创建的对象;
      2. 在匿名内部类中重写run方法
         1. 输出: 参加运动会,奔跑吧

## 训练案例6

### 训练考核知识点

接口

接口名作为形式参数类型

匿名内部类

### 训练描述

请用代码描述：

某高校举成立一个大球俱乐部,只要是会打篮球,踢足球,打排球的可以参加.

要求: 使用到接口,接口作为参数,匿名内部类.

提示: 把打篮球,踢足球,打排球功能抽取到接口中 ,在测试类中提供加入俱乐部的静态方法,参数为接口类;

### 操作步骤描述

1. 定义接口(Player)
   1. 抽象方法: 打篮球(playBasketball),踢足球(playFootball),打排球(playVolleyball)
2. 定义测试类Test
   1. 定义静态方法:进入运行会(join(Player p)),在方法中调用p的 打篮球,踢足球,打排球 方法
   2. 提供main方法
   3. 在main方法中
      1. 调用join(Player p),传入使用Player 接口的匿名内部类创建的匿名对象;
      2. 在匿名内部类中
         1. 重写playBasketball()方法
            1. 输出: 在打篮球
         2. .重写playFootball()方法
            1. 输出: 在踢足球
         3. .重写playVolleyball()方法
            1. 输出: 在打排球

# 关卡3

## 训练案例1

### 训练考核知识点

静态代码块

构造方法

成员变量

### 训练描述

看下面的程序说出执行结果并阐述程序的执行流程

**public** **class** Student {

**static** {

System.***out***.println("Student 类的静态代码块");

}

**public** Student(){

System.***out***.println("Student 类的构造方法");

}

**public** Student(String methodName){

System.***out***.println("在"+methodName+"创建Student对象调用的");

}

}

**public** **class** Test {

// 实例(对象)成员变量

Student s = **new** Student("成员变量显式初始化的时候");

// 静态代码块

**static** {

System.***out***.println("Test 类的静态代码块");

}

**public** **static** **void** main(String[] args) {

System.***out***.println("main方法");

// 创建学生对象

Student s1 = **new** Student("main方法中");

// 创建Test对象

Test t = **new** Test();

}

}

提示: 1. 类只有被使用就会加载到内存中

2. 只有创建对象才会为成员变量分配存储空间才能对成员变量赋值

## 训练案例2

### 训练考核知识点

抽象类

匿名内部类

### 训练描述

请使用代码描述:

某人养了几只宠物,每个宠物都有名字,每一个宠物都会吃,但是每个宠物吃的东西都不一样;宠物1叫tom吃鱼,宠物2叫小黑,啃骨头.

要求: 用到抽象类和匿名内部类

## 训练案例3

### 训练考核知识点

接口

接口作为形式参数

匿名内部类

### 训练描述

请使用代码描述

某公司成立一个游泳俱乐部,只要是会游泳的都可以加入;

要求: 使用到接口,接口作为参数,匿名内部类.

提示: 把游泳功能抽取到接口中 ,在测试类中提供参加运动会的静态方法