

FY18_广州_高校个体_Java9班 (JavaSE核心技术) (参考答案)

第一部分：选择题 (本部分 6 道题，满分 24 分)

1、 【单选题】 答案： C
题目解析：

2、 【单选题】 答案： B
题目解析：

3、 【单选题】 答案： A
题目解析：

4、 【单选题】 答案： C
题目解析：

5、 【单选题】 答案： B
题目解析：

6、 【单选题】 答案： B

第二部分：程序填空 (本部分 7 道题，满分 28 分)

1、 【填空题】 答案： finally
exception
finished

2、 【填空题】 答案： parent
child
child

3、 【填空题】 答案： Over.amethod

4、 【填空题】 答案： 23

5、 【填空题】 答案： 39

6、 【填空题】 答案： 0 2 4
题目解析：

7、 【填空题】 答案： pst.setString(1,"Java");
题目解析：

或者pst.setString("bookName","Java")

第三部分：问答题（本部分 3 道题，满分 18 分）

1、【问答题】答案：

static可以修饰的元素：

属性、方法、代码块

1) 静态变量特点：

所有对象共享

在类被载入时创建，只要类存在，static变量就存在

两种方式访问：

直接访问：类名.属性；

实例化后访问：对象名.属性

2) 静态方法特点：

不需要实例化，可以直接访问

两种方式访问：

直接访问：类名.方法名()

实例化后访问：对象名.方法名()

static方法的作用：

简化方法的使用；

便于访问静态属性；

注意事项

静态方法里只能直接访问静态成员，而不能直接访问类中的非静态成员

静态方法中不能使用this、super关键字

静态方法不能被非静态方法覆盖，静态方法不能修饰构造器

3) 静态代码块：一个类中由static关键字修饰的，不包含在任何方法体中的代码块

当类被载入时，静态代码块被执行，且只被执行一次

静态块经常用来进行类属性的初始化

三点错一个 -2分

错误多，最低0分

题目解析：

7-36 7-37 7-38

2、【问答题】答案：ArrayList和Vector都是使用数组方式存储数据，此数组元素数大于实际存储的数据以便增加和插入元素，它们都允许直接按序号索引元素，但是插入元素要涉及数组元素移动等内存操作，所以索引数据快而插入数据慢，Vector由于使用了synchronized方法（线程安全），通常性能上较ArrayList差，而LinkedList使用双向链表实现存储，按序号索引数据需要进行前向或后向遍历，但是插入数据时只需要记录本项的前后项即可，所以插入速度较快。

3、【问答题】答案：

equals()方法是Object类的方法，在Object类中比较两个对象引用是否指向同一个对象，一般都被子类方法覆盖，不再比较引用的值 --2分

"=="是比较运算符，当比较基本数据类型时，比较两个数据值是否相等；当比较引用数据类型时，比较两个引用是否指向同一个对象。--2分

题目解析：

第四部分：组合题（本部分 1 道题，满分 15 分）

1、【组合题】

1) 【问答题】答案：

//Student 类

class Student{

String name;

int age;

Student(String name,int age){

this.age=age;

this.name=name;

}

Student(){}

public void setName(String name){

this.name=name;

}

```

    public String getName(){
        return name;
    }
    public void setAge(int age){
        this.age=age;
    }
    public int getAge(){
        return age;
    }
}

```

题目解析：

2) 【问答题】 答案：

```

//Undergraduate类
class Undergraduate extends Student{
    String specially;
    Undergraduate(String n,int a,String s){
        name = n;
        age = a;
        specially = s;
    }
    Undergraduate(){

    }

    public void setSpecially(String specially){
        this.specially = specially;
    }
    public String getSpecially(){
        return specially;
    }
}
//Postgraduate类
class Postgraduate extends Student{
    String studydirection;
    Postgraduate(String n,int a,String s){
        name = n;
        age = a;
        studydirection = s;
    }
    Postgraduate(){

    }
    public void setStudydirection(String studydirection){
        this.studydirection = studydirection;
    }
    public String getStudydirection(){
        return studydirection;
    }
}

```

题目解析：

3) 【问答题】 答案：

```

public class TestStudent{ //程序入口
    public static void main(String[] args) {
        Student stu1 = new Student("Tom",19);
        Student stu2 = new Student("Anna",20);
        Postgraduate stu3 = new Postgraduate("John",25,"Computer Science");
        Postgraduate stu4 = new Postgraduate("Lucy",24,"Chinese");
        Undergraduate stu5 = new Undergraduate("Ansen",21,"Math");
        Student[] data = {stu1,stu2,stu3,stu4,stu5}; //Student类的数组
        System.out.println("Name\tAge");
        System.out.println("-----");
        System.out.println(stu1.getName()+"\t"+stu1.getAge());
        System.out.println(stu2.getName()+"\t"+stu2.getAge());
        System.out.println(stu3.getName()+"\t"+stu3.getAge()+"\t"+stu3.getStudydirection());
        System.out.println(stu4.getName()+"\t"+stu4.getAge()+"\t"+stu4.getStudydirection());
        System.out.println(stu5.getName()+"\t"+stu5.getAge()+"\t"+stu5.getSpecially());
        System.out.println("-----");
        data[0].setName("汤姆");
        stu2.setName("约翰");
        for(int i=0;i<data.length;i++){
            System.out.println(data[i].getName());
        }
    }
}

```

```
}  
}
```

题目解析：

第五部分：编程题（本部分 1 道题，满分 15 分）

1、【编程题】（考生可选的语言：Java）答案：

```
package com.neuedu;  
public class TenthTreeFall {  
    static double height = 100;  
    static double distance = 100;  
    public static void main(String[] args) {  
        for(int i=1; i<10; i++) {  
            distance = distance + height;  
            height = height / 2;  
        }  
  
        System.out.println("路程: " + distance);  
        System.out.println("高度: " + height / 2);  
    }  
}
```

题目解析：