编程题：面向对象封装、继承

public class Pet {

private String name;

public Pet() {

}

public Pet (String name) {

this.name = name;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public void print() { //打印方法

System.out.println("\n名字："+name);

}

public class Dog extends Pet{

private String color;

public String getColor() {

return color;}

public void setColor(String color) {

this.color = color;

}

public Dog () {

}

public Dog (String name,String color) {

super(name);

this.color=color;

}

public void print() { //打印方法

System.out.println("\n名字："+super.getName()+"\n健康度："+color);

}

public static void main(String[] args) {

Pet p=new Pet("张三");

Dog p1=new Dog("小小","黑色");//多态成立的另一个条件是在创建子类时候必须使用父类new子类的方式。

p.print();

p1.print();}

2、JDBC

// 1、创建驱动获取连接

Class.forName("com.mysql.jdbc.Driver");

con=DriverManager.getConnection("jdbc:mysql://127.0.0.1/jsd","root","1234");

// 2、创建语句对象

st=con.createStatement();

// 3、执行SQL语句

String sql="select \* from dept\_xu";

rs=st.executeQuery(sql);

// 4、处理结果集

while(rs.next()) {

dept=new Dept();

dept.setDeptno(rs.getInt("deptno"));

dept.setDname(rs.getString("dname"));

dept.setLocation(rs.getString("location"));

list.add(dept);

}

3、线程同步

public class WaitAndNotifyDemo{

//图片是否下载完毕

public static boolean isFinish = false;

public static Object object = new Object();

public static void main(String[] args) {

//下载图片的线程

final Thread download = new Thread(){

public void run(){

System.out.println("download:开始下载图片");

for(int i=0;i<=10;i++){

System.out.println("download:已完成"+i+"%");

try {

Thread.sleep(50);

} catch (InterruptedException e) {

e.printStackTrace();

}

}

System.out.println("download:图片下载完毕");

isFinish = true;//表示图片下载完毕

synchronized(object){

//通知object身上的等待的线程解除阻塞

object.notify();

}

System.out.println("download:开始下载附件");

for(int i=0;i<=10;i++){

System.out.println("download:已完成"+i+"%");

try {

Thread.sleep(50);

} catch (InterruptedException e) {

e.printStackTrace();

}

}

System.out.println("download:附件下载完毕");

}

};

download.start();

}

}

4、集合

Set:

public static void main(String[] args) {

Set<String> set = new HashSet<String>();

set.add("one");

set.add("two");

set.add("three");

for(String str : set){

System.out.println(str);

}

}

public class HashMapStudent {

public static void main(String[] args) {

Map<String,List<Student>> map=new HashMap<String,List<Student>>();

List<Student> list=new ArrayList<>();

list.add(new Student(1, "张三"));

list.add(new Student(2,"李四"));

List<Student> list1=new ArrayList<>();

list1.add(new Student(1,"王五"));

list1.add(new Student(2, "王麻子"));

map.put("强化班",list);

map.put("基础班",list1);

//获取强化班的学生

List<Student> slist = map.get("强化班");

System.out.println("强化班的学生");

for(Student s : slist){

System.out.println(s);

}

//获取普通版的学生

List<Student> nlist = map.get("基础班");

System.out.println("普通班的学生");

for(Student s : nlist){

System.out.println(s);

}

}

}

class Student{

private int id;

private String name;

public Student(int id, String name) {

super();

this.id = id;

this.name = name;

}

@Override

public String toString() {

return "id=" + id + ", name=" + name;

}

5、IO

读文件：

public static void main(String[] args) throws Exception{

FileReader reader =

new FileReader("fx.txt");

int c = -1;

while((c=reader.read())!=-1){

System.out.println((char)c);

}

reader.close();

}

写文件：

public static void main(String[] args) throws Exception{

FileWriter writer =

new FileWriter("fx.txt");

BufferedWriter bw =

new BufferedWriter(writer);

bw.write("呵呵呵");

bw.close();

}