**Exercise1：**

**Person:**

**public class Person {  
 private String name;  
 private int age;  
  
 public Person(){  
 this.name="John";  
 this.age=18;  
 }  
  
 public Person(String name,int age){  
 this.name=name;  
 this.age=age;  
 }  
  
 public int getAge(){  
 return this.age;  
 }  
  
 public String getName(){  
 return this.name;  
 }  
  
 public void setAge(int age){  
 this.age=age;  
 }  
  
 public void setName(String name){  
 this.name=name;  
 }  
  
 public String enjoy(){  
 return "being with nature";  
 }  
  
 public String enjoy(String a){  
 return a;  
 }  
  
 //Override  
 public String toString(){  
 return "i am a person, my name is "+name+", age is "+age+". i enjoy with "+enjoy();  
 }  
}**

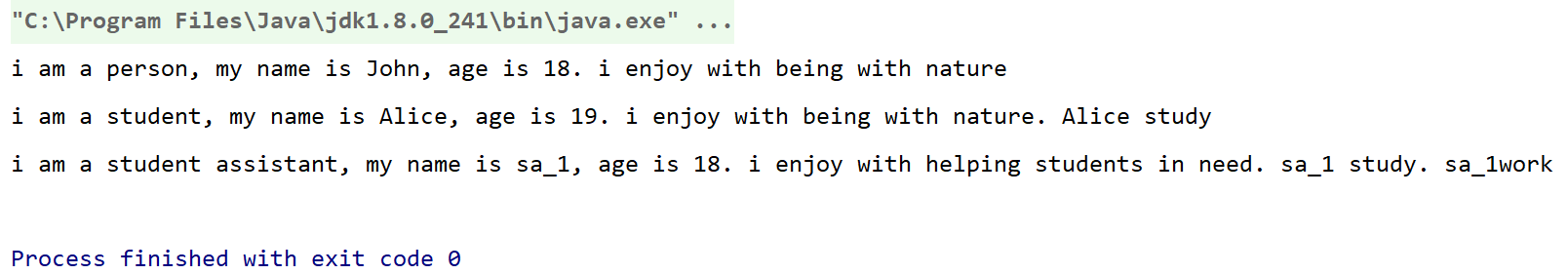
**Student:  
public class Student extends Person{  
 private String classStatus;  
  
 public Student(){  
 classStatus="student";  
 }  
  
 public Student(String name,int age,String classStatus){  
 super(name,age);  
 this.classStatus=classStatus;  
 }  
  
 public String getClassStatus(){  
 return classStatus;  
 }  
  
 public void setClassStatus(String classStatus){  
 this.classStatus=classStatus;  
 }  
  
 public String study(){  
 return getName()+" study";  
 }  
  
 //override  
 public String toString(){  
 return "i am a "+classStatus+", my name is "+super.getName()+", age is "+super.getAge()+". i enjoy with "+enjoy();  
 }  
}**

**StudentAssistant:**

**public class StudentAssistant extends Student{  
 static int id=0;  
  
 public StudentAssistant(){  
 id++;  
 setName("sa\_"+id);  
 }  
  
 //override enjoy()  
 public String enjoy(){  
 return "helping students in need";  
 }  
  
 public String work(){  
 return getName()+"work";  
 }  
  
 public String toString(){  
 return "i am a student assistant, my name is "+getName()+", age is "+getAge()+". i enjoy with "+enjoy();  
 }  
}**

**TestInherit:**

**public class TestInherit {  
 public static void main(String[] args) {  
 Person p1=new Person();  
 Student s1=new Student("Alice",19,"student");  
 StudentAssistant sa1=new StudentAssistant();  
  
 System.out.println(p1);  
 System.out.println(s1+". "+s1.study());  
 System.out.println(sa1+". "+sa1.study()+". "+sa1.work());  
 }  
}**



**Exercise2：**

**public class TestPolymorphism {  
 public static void main(String[] args) {  
 Person p1=new Person();  
 Student s1=new Student();  
 StudentAssistant sa1=new StudentAssistant();  
  
 Person p=new Person("pp",15);  
 introduction(p,"");  
 p=new Student("ss",17,"Student");  
 Student ps=(Student)p;  
 introduction(ps,". "+ps.study());  
 p=new StudentAssistant();  
 StudentAssistant pss=(StudentAssistant)p;  
 introduction(pss,". "+pss.study()+". "+pss.work());  
 }  
  
 public static void introduction(Person p,String x){  
 System.out.println(p+x);  
 }  
}**

