

«باسمه تعالی»

تمرین سری پنجم

سوال اول)

- 1) A method that is declared **private** cannot be overridden in a subclass.
- 2) Can hold objects of more than one type: it is the definition of **polymorphism**.
- 3) Objects of **abstract** super classes cannot be instantiated.
- 4) All classes in Java inherit directly or indirectly from the **object** class.
- 5) methods declared in an interface are by default **abstract**.
- 6) At interfaces all fields are public, **static** and **final**.
- 7) In Java we can only use multiple inheritance for **interfaces**.

سوال دوم)

- 1) When you instantiate a sub class, super class constructor will be also executed.

**True**

- 2) An abstract class has no use until unless it is extended by some other class.

**True**

- 3) A constructor of abstract class is called when an instance of an inherited class is created.

**True**

- 4) Abstract classes may contain non-final variables, whereas variables in interface are final, public and static.

**True**

5) Protected members are accessible within a package and inherited classes outside the package.

True

سوال سوم)

### **:SubType and SuperType**

می دانیم براساس تعریف هر کلاس یک type را مشخص میکند. در جاوا در روابط ارث بری می دانیم به کلاس پدر superclass و به کلاس های فرزند subclass گفته میشود. هر عضوی که داخل superclass تعریف شود را supertype و هر عضوی که داخل subclass باشد را subtype گویند. به نحوی میتوان گفت subtype ها زیر مجموعه supertype ها هستند.

### **:Diamond Problem**

برای مثال فرض کنیم کلاس D از دو کلاس به نام های B و C ارث میبرد و این دو کلاس خودشان از کلاس A ارث میبرند. حال اگر متدی داخل کلاس A باشد که B و C آن را override کرده باشند ولی D نکرده باشد مشکلی که پیش می آید این است که D کدام ورزن از این متد را به ارث میبرد؟ آن متدی که داخل B اورراید شده است یا متدی که در C آمده است؟ پس همانطور که مشخص است نمیتوان تشخیص داد که به این مشکل Diamond Problem گویند.

سوال چهارم)

قسمت الف)

**Food f1= new Pasta();**

مجاز میباشد. از آنجایی که طبق تعریف متغیر f1 یک supertype میباشد پس میتوان شی جدیدی از subtype را در آن قرار داد برای توضیح بیشتر میتوان گفت static type متغیر f1 از جنس Food است و dynamic type آن از جنس پاستا میباشد و با توجه به ارث بری پاستا از فود این کار امکان پذیر است.

**Food f2=new Alfredo();**

مجاز میباشد. همانطور که گفته شد طبق polymorphism میتوان اشیای از جنس subtype را داخل supertype نگه داشت. داد برای توضیح بیشتر میتوان گفت static type متغیر f2 از جنس Food است و dynamic type آن از جنس آلفردو میباشد و با توجه به ارث بری غیر مستقیم آلفردو از فود این کار امکان پذیر است.

**Pizza pi1=new Food();**

غیرمجاز است. زیرا Food یک superclass میباشد و اشیای از جنس Super type را نمیتوان داخل subtype قرار داد.

**Pasta pa1=new Alfredo();**

مجاز میباشد. کلاس Alfredo به صورت مستقیم از کلاس Pasta ارث بری میکند بنابراین میتوان شی ای از جنس subtype را داخل متغیری از جنس supertype نگه داشت. برای توضیح بیشتر میتوان گفت static type متغیر pa1 از جنس Pasta است و dynamic type آن از جنس آلفردو میباشد و با توجه به ارث بری مستقیم آلفردو از پاستا این کار امکان پذیر است.

**Pizza pi2=new Pepperoni();**

مجاز میباشد. کلاس Pepperoni به صورت مستقیم از کلاس Pizza ارث بری میکند بنابراین همانند قبل طبق polymorphism میتوان شی ای از جنس subtype را داخل supertype نگه داشت.

قسمت ب)

**f1=pe1;**

مجاز میباشد. زیرا متغیری از جنس subtype را داخل متغیری از جنس supertype قرار داده ایم. درواقع f1 یک متغیر با static type ای از جنس Food که یک Superclass میباشد است و pe1 یک شی ای از subclass میباشد.

**pi1=pe1;**

مجاز میباشد. چون کلاس Pepperoni از کلاس Pizza ارث بری میکند بنابراین در این جا نیز polymorphism رعایت شده و میتوان یک subtype را داخل super type قرار داد. درواقع pi1 یک متغیر با static type ای از جنس Pizza که یک Superclass میباشد است و pe1 یک شی از subclass میباشد

**a1=f1;**

غیرمجاز است. Static type متغیر f1 از جنس کلاس Food میباشد و static type متغیر a1 از جنس کلاس Alfredo میباشد و Alfredo از Food ارث بری میکند بنابراین نمیتوان یک super type را داخل subtype قرار داد.

**f2=pi1;**

مجاز میباشد. زیرا متغیری از جنس subtype را داخل متغیری از جنس supertype قرار داده ایم. درواقع f2 یک متغیر با static type ای از جنس Food که یک Superclass میباشد است و pi1 یک شی از subclass میباشد. بنابراین polymorphism رعایت شده است

**f1=pe1;**

مجاز میباشد. زیرا متغیری از جنس subtype را داخل متغیری از جنس supertype قرار داده ایم. درواقع f1 یک متغیر با static type ای از جنس Food که یک Superclass میباشد است و pe1 یک شی از subclass میباشد