

سوال اول

- A) protected
- B) constructor
- C) super
- D) Object

سوال دوم

- (A) درسته ديگه چون اين رابطه يك طرفه است و فقط subclass ها ميتوانند از superclass ها استفاده كنند.
- (B) غلطه...آخه override درستشه.
- (C) اينم درسته

سوال سوم

Polymorphism: انگار يعني يك كار رو بشه با توجه به شرايط موجود تصميم گيري بشه كه چجوري انجام بشه. polymorphism انواعي دارد كه با overloading قبل آشنا شديم.

Overriding: يك نوع polymorphism است به صورتي كه در subclass و superclass متد همنامي داريم و در حين اجرا JVM تصميم ميگيرد كه از کدام استفاده كند.

Substitution: فرض ميكنيم كه A يك Superclass است و B هم subclass است. Substitution ميگه كه اين B ميتونه مساوي A قرار بگيره چون در واقعيت هم B نوعي از A است.

Abstract Class: اگر در يك كلاس ، متدي به حالت abstract (كه يعني فقط يه چيزي مثل prototype از اون متد رو براي معرفي مينويسيم) داشته باشيم ، كلاس ما به حالت abstract در مي آيد.

Interface: نوعي كلاس هستش كه همه ي متد هاي آن abstract هستند و هيچ constructor اي ندارد و براي موقعي كه يك كلاس از ۲ كلاس ارث بري دارد ، استفاده ميشود.

سوال چهارم

(الف)

- (i) درست.درسته چون TextBook از فرزندان Book محسوب ميشه.
- (ii) درست.اينم درسته چون Mystery فرزند Novel و اون هم فرزند Book هست.
- (iii) غلط.غلطه چون توي Novel ، چيز هايي هست كه با new كردن Book اونا معين نميشوند.
- (iv) درست.اين درسته دليلش هم اينه كه Mystery فرزند Novel ميباشد.
- (v) غلط.غلطه دليلش مثل مورد iii ميباشد.

(ب)

- (i) درست.درسته چون ما ميتونيم sub-type رو assign كنيم به super-type.
- (ii) غلط.نه اين غلطه ، براي اينكار بايد از casting استفاده بشه.
- (iii) درست.اينم درسته دليلش هم مثل i.
- (iv) غلط.نه نه اين كاملا غلطه چون دو نوع sub-type رو برابر هم قرار داده.
- (v) درست.اينم درسته دليلش هم مثل i.
- (vi) غلط. غلطه دليلش هم همانند دليل ii هست.