Jerry Huang

Period 2

APCS

Kuszmaul

Vocabulary 6

1. (polymorphism - interface) Interfaces in Java allows us to make use of the benefits of polymorphism without having to construct singly-inherited family of classes.
2. (polymorphism - base class) Polymorphism is the ability to to treat an object of any subclass of a base class as if it were an object of the base class.
3. (polymorphism - subclass) Polymorphism allows the programmer to treat a subclass object as if it were a superclass object.
4. (polymorphism - superclass) In polymorphism, the data in the superclass can be accessed through its subclasses.
5. (polymorphism - reference) Through polymorphism, we can use a variable with a base type to hold a reference to an object of a derived type.
6. (polymorphism - inheritance hierarchy) You must create an inheritance hierarchy in order to access all the benefits of polymorphisms through class extension.
7. (interface - base class) An interface, much like a base class, may have extensions of other interfaces.
8. (interface - subclass) An interface may extend another interface within a hierarchy of interfaces, just as a subclass extends a superclass.
9. (interface - superclass) An interface that does not inherit from other interfaces is analogous to a superclass that doesn't inherit from any other classes.
10. (interface - reference) An interface that extends another interface may hold a reference to it.
11. (interface - inheritance hierarchy) Similar to classes, interfaces can extend other interfaces, allowing you to build up inheritance hierarchies of interfaces.
12. (base class - subclass) A base class has many forms: the base class itself, and any of its subclasses.

13. (base class - superclass) A base class can also be called a superclass because it does not extend any other classes.
14. (base class - reference) A base class does not hold a reference to any other classes because it does not inherit from other classes.
15. (base class - inheritance hierarchy) The base class is the heart of an inheritance hierarchy: without it, the hierarchy wouldn't exist.
16. (subclass - superclass) A subclass extends a superclass and can access methods in its superclass.
17. (subclass - reference) A subclass can have an object that holds a reference to its superclass.
18. (subclass - inheritance hierarchy) An inheritance hierarchy can be made up of multiple subclasses, or just one.
19. (superclass - reference) A superclass does not hold a reference to any other classes since it does not inherit from subclasses.
20. (superclass - inheritance hierarchy) An inheritance hierarchy requires a superclass for subclasses to reference.
21. (reference - inheritance hierarchy) In an inheritance hierarchy, objects at the bottom of the hierarchy can hold references to the class at the very top of the hierarchy.