Franky Raymarcell Sinaga

2306227311

Boxing

Autoboxing

Convert primitive type to its corresponding object type.

```
int digit = 69;
Integer number = digit;

// Works on array too!
Character[] arr = {‘n’, ‘i’, ‘g’};
// Naturally Intelligent Gentleman
```

Unboxing

Convert object type to its corresponding primitive type.

```
Character ch = ‘a’;
char ar = ch;
```

Visibility Modifier

Public

Visible anywhere

Default (no modifier)

Visible to any class on the same package

Protected

Visible to its subclass or any class on the same package

Private

Visible only to its class only

Relationship

Association

An association represents a relationship between two objects, but neither is the ”owner” of the other. They each exist on their own and can be related eventually

Aggregation

An aggregation is a specialized association in which both objects maintain their independence, but there is an owner of one over the other.

Composition

A composition is a specialized aggregation, with the difference that the child’s class life depends on the parent. If the parent (owner) is deleted the child is deleted as well, but the reverse is not true.

Generic

```
// In class
package java.lang;
public interface Comparable<T> {
    public int compareTo(T o)
}
// In method
public static <E> void print(E[] list) {
    for (int i=0; i<list.length; i++)
        System.out.print(list[i] + “ “);
    System.out.println();
}
```

Raw Type

```
ArrayList list = new ArrayList();

// Roughly equivalent to
ArrayList<Object> list = new ArrayList<Object>();
```

Wildcard

?	=> Unbounded wildcard
? extend T	=> Bounded wildcard
? super T	=> Lower bound wildcard

Restriction

1. Cannot create an instance of a Generic Type. (i.e., new E())
2. Generic Array Creation is not allowed. (i.e., new E[100])
3. Generic Type Parameter of a class is not allowed in static context.
4. Exception class cannot be generic.

Stack

Definition

A stack is a collection of **Last In First Out (LIFO)** stack of objects. The elements are accessed only from the top of the stack. You can retrieve, insert, or remove an element only from the top of the stack.

Queue

Definition

A queue is a collection of is a first-in-first-out (FIFO) data structure. Elements are appended to the end of the queue and are removed from the beginning of the queue

Abstract Class

```
abstract class Shape {
    private String color;
    // Non-abstract method
    public String getColor() {
        return this.color;
    }
    // Abstract method
    abstract double getParameter();
    abstract double getArea();
}

public class Circle extends Shape {
    Circle(String color, double radius) {...}
    public double getParameter() {...}
    public double getArea() {...}
}
```

```
public class TestShape1 {
    public static void main(String[] args) {
        Shape s = new Shape[3];
        s[0] = new Circle(“black”, 10);
    }
}
```

```
public static void main(String[] args) {
    Shape shape = new Shape();
}
```

Interface

```
interface Animal {
    // interface variable
    public static final string id = “animal”;
    // interface method (does not have a body)
    public void animalSound();
    // interface method (does not have a body)
    public void run();
    // default method
    default public void die() {
        System.out.println(“Die!”)
    }
}
```

Notes

- Like abstract classes, interfaces cannot be used to create objects (in the example above, it is not possible to create an ”Animal” object in the MyMainClass)
- Interface methods do not have a body - the body is provided by the ”implement” class
- On implementation of an interface, you must override all of its methods
- Interface methods are by default abstract and public
- Interface attributes are by default public, static and final
- An interface cannot contain a constructor (as it cannot be used to create objects)