Recommended Reading

January 26, 2000

• of Unreal. Look at Programming Languages by Tim Sweeney, Principle architect

class only if you want to express "non-existence". Assign null to a variable of some
null represents "non-existence". It is automatically assigned the value null.

When a variable of some class is declared without any initial instantiation,

What is null?

January 26, 2000
Rectangle

- `width` and `height` are said to be instance fields (or variables) of `Rectangle`

- `width` and `height` of `t`, `s`, and `rectangle` of `s`. It does not know anything about the `width` and `height` of `t`. The code for `rectangle` is executed and can only access the `width` and `height` of `s`.

```java
s.darea();
```

```java
Rectangle t = new Rectangle(3, 4);
```

```java
Rectangle s = new Rectangle(6, 7);
```

Instance Variables and Instance Methods
• It can only be called on an existing instance of a class.

double area() is said to be an instance method of a rectangle.

Instance Variables and Instance Methods (cont.)
the keyword static.

and that can be accessed by all methods in Rectangle. In Java, we use

Answer: a field that is unique and global to all instances of Rectangle

during the course of our program. What do we need?

Suppose we want to keep track of how many rectangles are being created

Static Variables and Static Methods

January 26, 2000
{  
    ... other code //
    -jcount++;
    -jcount is incremented by 1.  
    -jcount is incremented by 1. //
    Each time this constructor is called, //

    }  

    public Rectangle(double width, double height)  
    
    ... other code //

    private static int -jcount; // Initial value is 0.

    }  

    class Rectangle extends Shape

    Static Variables and Static Methods (cont.)
System.out.println(clipPizza);

Example: System is a class. out is a static variable.

(Recall main.)

- Static methods can be called before any instantiation of the class.
- Static fields.
- Static methods cannot access instance fields. They can only access

Static Variables and Static Methods (cont.)
subclasses), namely rectangle and circle. UML diagram shows the taxonomy tree of AShape and its variants (or A class and all of its subclasses form a taxonomy tree. The above
A variable of class AShape can be assigned any instance of subclasses of AShape at any time in a program. AShape is said to be polymorphic.

```java
Circle u = new Rectangle(5, 6); // OK
```

```java
t = s; // OK, the old Rectangle is gone.
```

```java
AShape t = new Rectangle(3, 4); // OK
```

```java
AShape s = new Circle(2.7); // OK
```

Polymorphism
We can think of polymorphism as viewing the taxonomy tree from the top down.

Class can be represented any of its subclasses. Polymorphism means a class can be represented any of its subclasses, but not the other way around. In general, a variable of a superclass can be assigned an instance of any subclass.
We can think of inheritance as viewing the taxonomy tree from the bottom up.

Inheritance

January 26, 2000

Comp 212