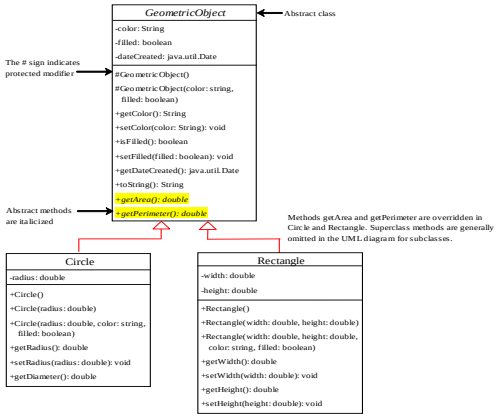
Lecture 12
Abstract Classes

Abstract Classes and Abstract Methods



abstract method in abstract class

- An abstract method cannot be contained in a non-abstract class.
- If a subclass of an abstract superclass does not implement all the abstract methods, the subclass must be defined abstract.
- In other words, in a non-abstract subclass extended from an abstract class, all the abstract methods must be implemented, even if they are not used in the subclass.

object cannot be created from abstract class

- An abstract class cannot be instantiated using the new operator, but you can still define its constructors, which are invoked in the constructors of its subclasses.
- For instance, the constructors of GeometricObject are invoked in the Circle class and the Rectangle class.

abstract class without abstract method

- A class that contains abstract methods must be abstract.
- However, it is possible to define an abstract class that contains no abstract methods.
- In this case, you cannot create instances of the class using the new operator. This class is used as a base class for defining a new subclass.

5

superclass of abstract class may be concrete

- A subclass can be abstract even if its superclass is concrete.
- For example, the Object class is concrete, but its subclasses, such as GeometricObject, may be abstract.

6

concrete method overridden to be abstract

A subclass can override a method from its superclass to define it abstract. This is rare, but useful when the implementation of the method in the superclass becomes invalid in the subclass. In this case, the subclass must be defined as abstract.

7

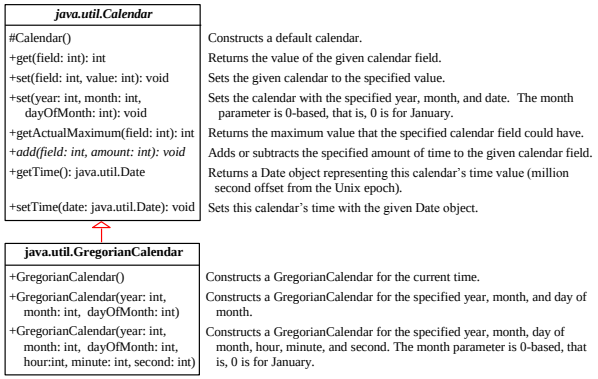
abstract class as type

- You cannot create an instance from an abstract class using the new operator, but an abstract class can be used as a data type.
- Therefore, the following statement, which creates an array whose elements are of GeometricObject type, is correct.

```
GeometricObject[] geo = new    GeometricObject[10];
```

8

The Abstract Calendar Class and Its GregorianCalendar subclass



9

The Abstract Calendar Class and Its GregorianCalendar subclass

- An instance of java.util.Date represents a specific instant in time with millisecond precision.
- java.util.Calendar is an abstract base class for extracting detailed information such as year, month, date, hour, minute and second from a Date object.
- Subclasses of Calendar can implement specific calendar systems such as Gregorian calendar, Lunar Calendar and Jewish calendar.
- Currently, java.util.GregorianCalendar for the Gregorian calendar is supported in the Java API.

10

The GregorianCalendar Class

You can use new GregorianCalendar() to construct a default GregorianCalendar with the current time and use new GregorianCalendar(year, month, date) to construct a GregorianCalendar with the specified year, month, and date. The month parameter is 0-based, i.e., 0 is for January.

11