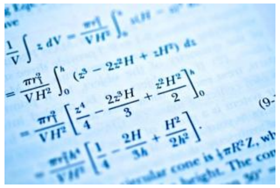
Abstract Classes

EHSDI
eBuzima

Introduction

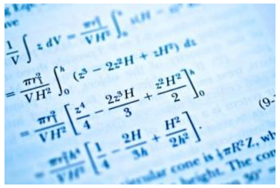


- Inheritance is a very useful feature of Java because it allows us to extract common functionality from classes into super-classes
- For example, a Shape class may contain the functionality common to different shape classes, and we can write...

```
Shape s1 = new Triangle();
Shape s2 = new Circle();
Shape s3 = new Octagon();
```

EHSDI
eBuzima

Introduction



- Different subclass objects can even be stored in the same static array or collection, e.g.

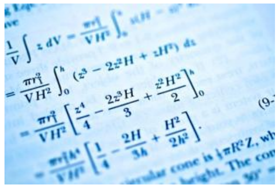
```
Shape[] myShapes = new Shape[2];
myShapes[0] = new Triangle();
myShapes[1] = new Circle();
```

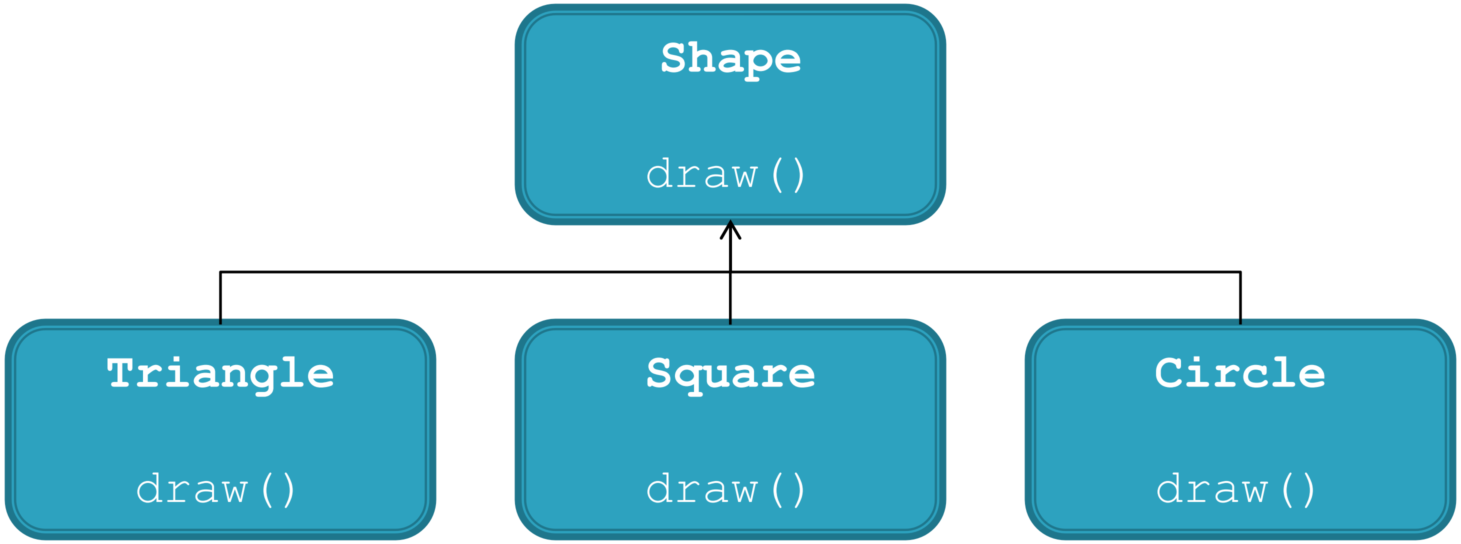
or

```
ArrayList<Shape> myList = new ArrayList<Shape>();
myList.add(new Triangle());
myList.add(new Circle());
```

EHSDI
eBuzima

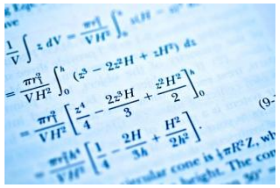
Example

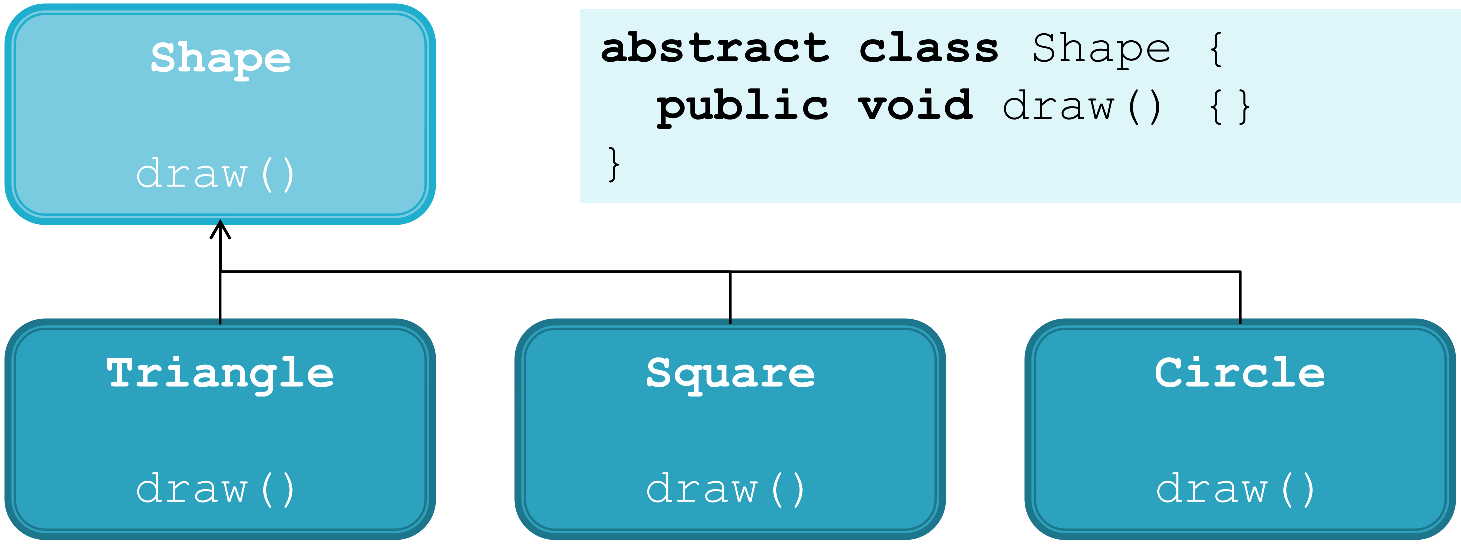




- How would one implement the draw method in the shape class?
- Should the shape class ever be instantiated?

Example





```
abstract class Shape {
  public void draw() {}
}
```

- Declaring the class as `abstract` prevents it from instantiated, i.e.

```
Shape s = new Shape();
```

Compiler error

EHSDI
eBuzima

Abstract methods

- Methods can also be declared `abstract`
- This means that they don't have an implementation
- Subclasses MUST provide an implementation or be abstract themselves
- A class with any abstract methods must be abstract itself, e.g.

```
abstract class Shape {
  public abstract void draw();
}
```

Abstract methods

```
class Shape {
  public abstract void draw();
}
```

Wrong: A class with abstract methods must be abstract

```
abstract class Shape {
  public abstract void draw();

  public String toString() {
    return "Shape";
  }
}
```

OK: An abstract class can have a mixture of abstract and normal methods

Abstract methods

```
abstract class Shape {
  public abstract void draw();
}
```

```
class Square extends Shape {
  public void draw() {
    ...
  }
}
```

```
abstract class Polygon extends Shape {
}
```

OK: A subclass can implement all abstract methods

OK: A subclass doesn't have to implement a method if it is abstract as well

References

- Sun's Java Tutorials: http://java.sun.com/docs/books/tutorial/java/IandI/abstract.html

EHSDI eBuzima