```java
abstract class Beverage {
  public String description = "Unknown Beverage";

  public String getDescription() {
    return description;
  }

  public abstract double cost();
}

class Espresso extends Beverage {
  public Espresso() {
    this.description = "Espresso";
  }

  public double cost() {
    return 1;
  }
}

class HouseBlend extends Beverage {
  public HouseBlend() {
    this.description = "House Blend";
  }

  public double cost() {
    return 1.5;
  }
}

abstract class Condiment extends Beverage {
  public abstract String getDescription();
}

class Mocha extends Condiment {
  private final Beverage beverage;

  public Mocha(Beverage beverage) {
    this.beverage = beverage;
  }

  public double cost() {
    return 0.20 + beverage.cost();
  }

  public String getDescription() {
    return this.beverage.getDescription() + ",Mocha";
  }
}

class Whip extends Condiment {
  private final Beverage beverage;

  public Whip(Beverage beverage) {
    this.beverage = beverage;
  }

  public double cost() {
    return 0.15 + beverage.cost();
  }

  public String getDescription() {
    return this.beverage.getDescription() + ",Whip";
  }

  class client {
    public static void main(String[] args) {
      Beverage b = new Mocha(new Mocha(new Whip(new HouseBlend())));
      System.out.println(b.getDescription());
      System.out.println(b.cost());
    }
  }

}
```

Handwritten annotation (circle around Mocha class, lines 35–49):

4 3 2 2
Hb, k, m, m

Pattern STRUTTURALI	Pattern CREAZIONALI	Pattern COMPORTAMENTALI
☐ Adapter ☑ Decorator ☐ Composite	☐ Abstract Factory ☐ Singleton	☐ Observer ☐ State ☐ Visitor ☐ Strategy