6.6 Enumeration as a Class

- Consider defining an enumeration for suits of cards

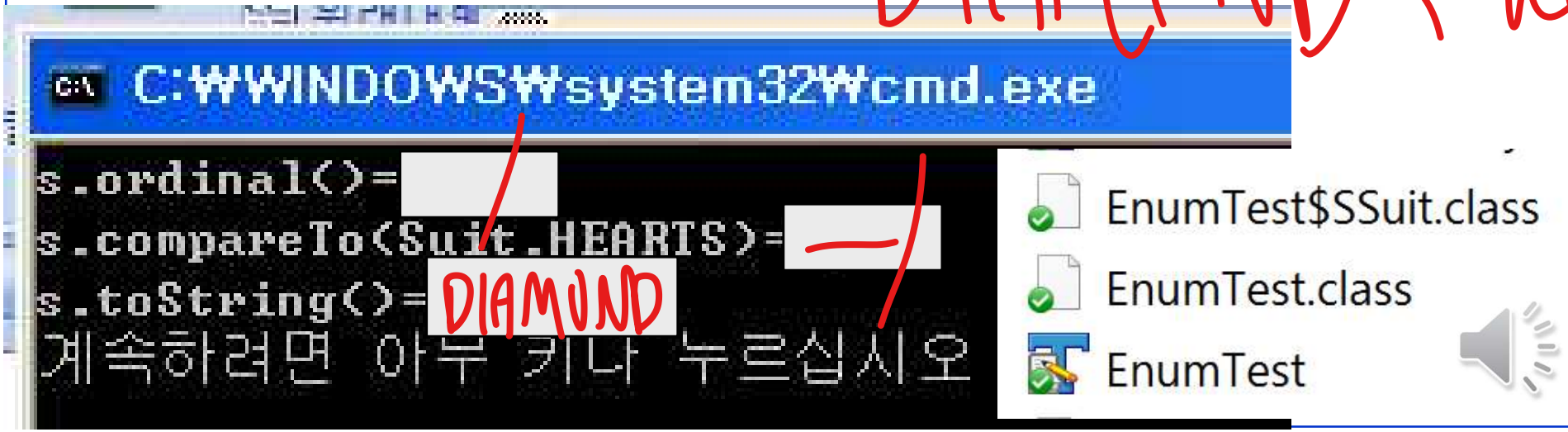
  ```
  enum Suit {CLUBS, DIAMONDS, HEARTS, SPADES}
  ```

- Compiler creates  with methods
  - » `equals`
  - » `compareTo`
  - » `ordinal`
  - » `toString`
  - » `valueOf`

```
/** An enumeration of card suits. */
public class EnumTest
{
   enum SSuit {CLUBS, DIAMONDS, HEARTS, SPADES}
    public static void main(String[] args)
    {
                    SSuit s = SSuit.DIAMONDS;
                    System.out.println("s.ordinal()="+s.ordinal());

          System.out.println("s.compareTo(SSuit.HEARTS)="+s.compareTo(
SSuit.HEARTS));
                    System.out.println("s.toString()="+s.toString());
          }
}
```

DIAMOND < HEART



Enumeration as a Class

- View enhanced enumeration, listing 6.20
  `enum Suit`
- Note
  - » Instance variables
  - » Additional methods
  - » Constructor

// Listing 6.20

```
/** An enumeration of card suits. */
enum Suit
{
    CLUBS ("black"), DIAMONDS ("red"), HEARTS ("red"),
        SPADES ("black");

    private final String color;

    private Suit (String suitColor)
    {
        color = suitColor;
    }


    public String getColor ()
    {
        return color;
    }
}
```

```
/** An enumeration of card suits. */
public class EnumTest02
{
   public static void main(String[] args)
   {
                  Suit s = Suit.DIAMONDS;
                  System.out.println("s.ordinal()="+s.ordinal());

          System.out.println("s.compareTo(Suit.HEARTS)="+s.compareTo(S
uit.HEARTS));
                  System.out.println("s.toString()="+s.toString());
                  System.out.println("s.getColor() = "+s.getColor());
          }
}
```