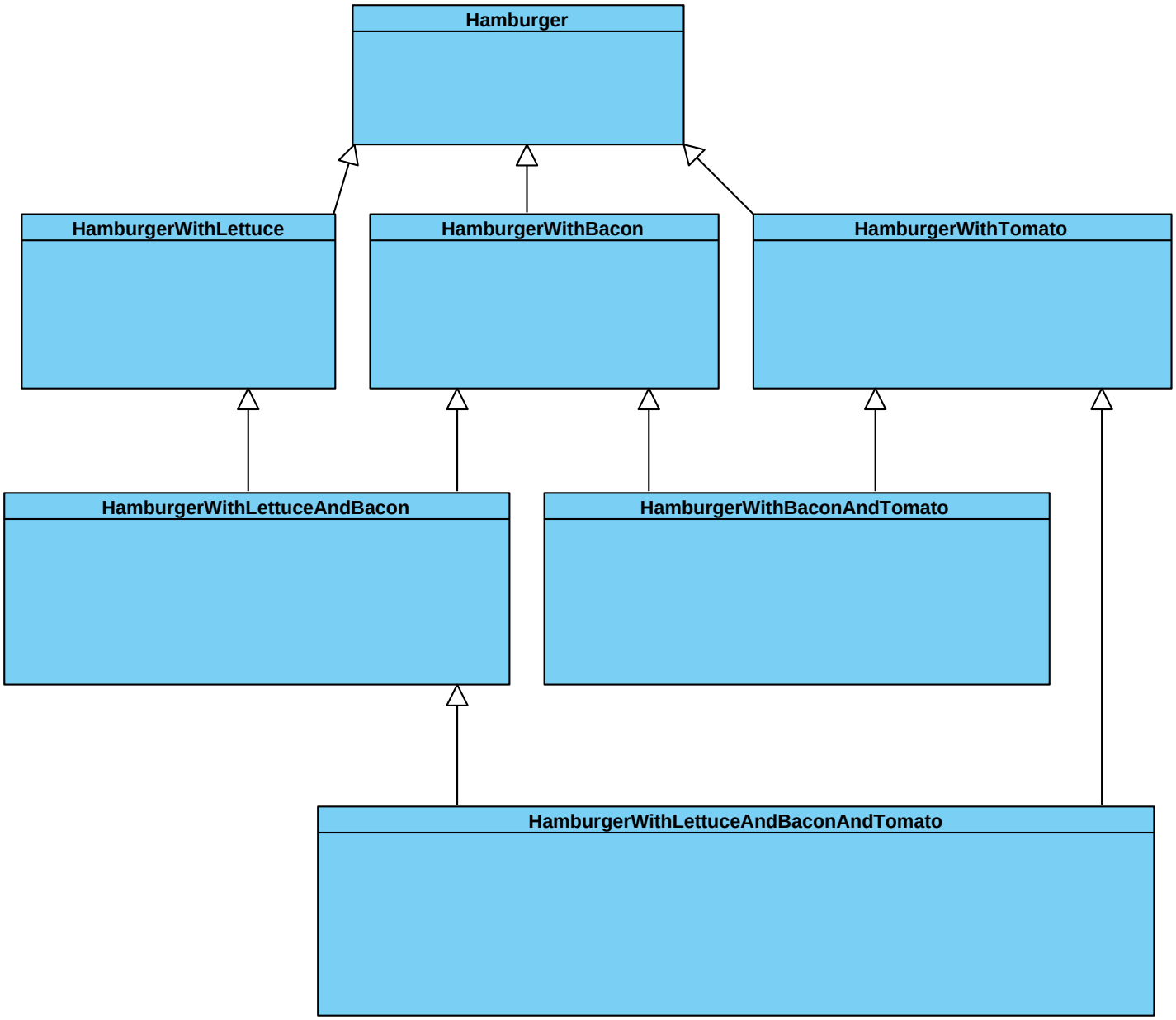
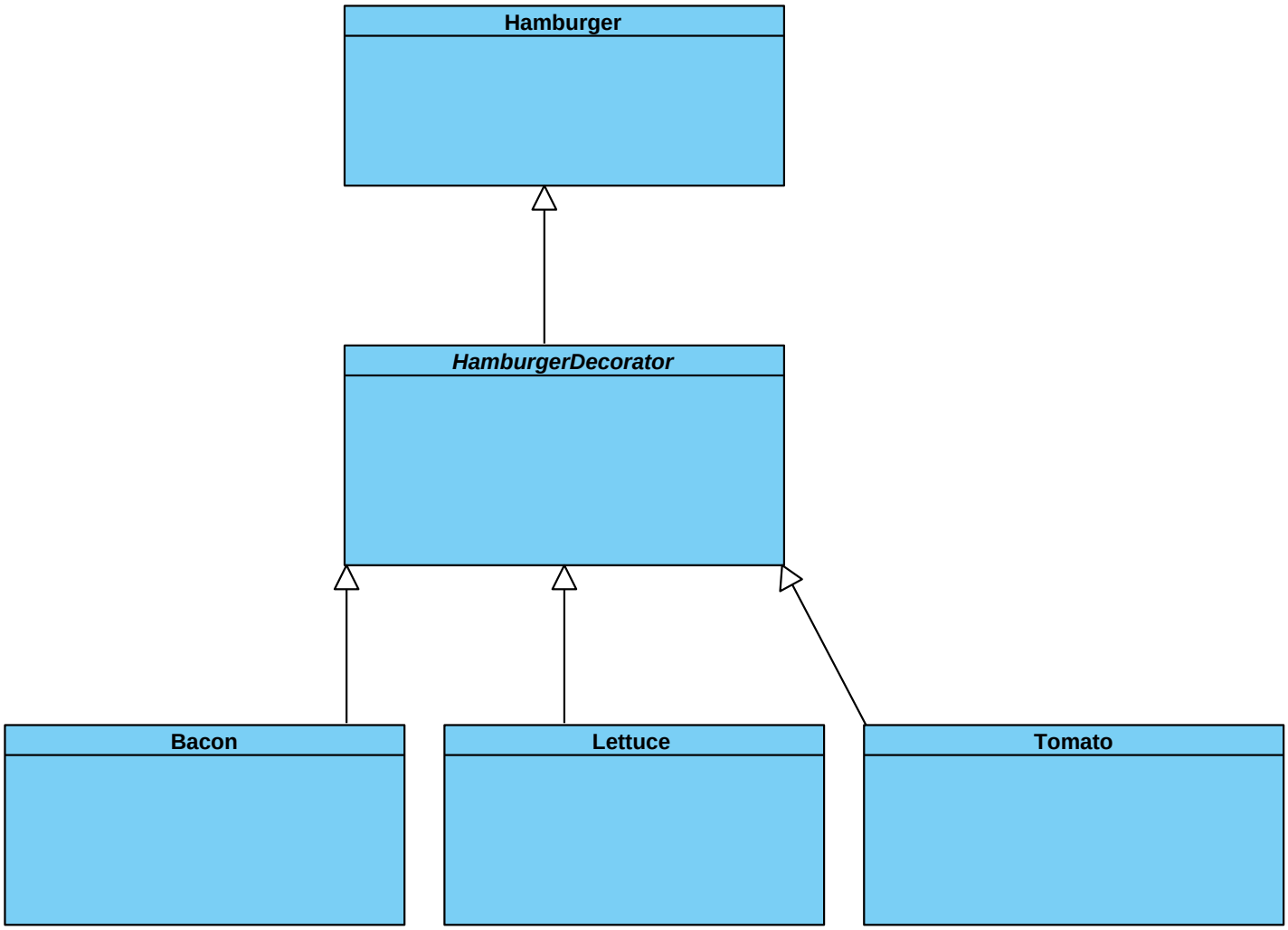
Decorator

Francis Ani



Subclassing Nightmare



Decorator Pattern

```
 // Hamburger.java (the “common denominator” class)
public class Hamburger
{
	public Hamburger()
	{
	}

	public String description()
	{
		return "hamburger ";
	}

	public String eat()
	{
		return "Tasty! "; // hypothetical reaction
	}


}
```

```
// HamburgerDecorator.java  (the abstract decorator class)

public abstract class HamburgerDecorator extends Hamburger
{
   public abstract String description();
   public abstract String eat();
}
```

```
// Bacon.java  (a concrete decorator)
public class Bacon extends HamburgerDecorator
{
	Hamburger hamburger;

	public Bacon(Hamburger h){
		hamburger = h;
	}
	public String description(){
		return hamburger.description() + " with bacon";
	}
	public String sizzle(){
		return "Sizzling! ";
	}
	public String eat(){
		return hamburger.eat() + sizzle();
	}
}
```

```
// Lettuce.java (a concrete decorator)
public class Lettuce extends HamburgerDecorator
{
	Hamburger hamburger;

	public Lettuce(Hamburger h){
		hamburger = h;
	}
	public String description(){
		return hamburger.description() + " with lettuce";
	}
	public String crunch(){
		return "Crunchy! ";
	}
	public String eat(){
		return hamburger.eat() + crunch();
	}
}
```

```
// Tomato.java (a concrete decorator)
public class Tomato extends HamburgerDecorator
{
	Hamburger hamburger;

	public Tomato(Hamburger h){
		hamburger = h;
	}
	public String description(){
		return hamburger.description() + " with tomato";
	}
	public String drip(){
		return "Juicy! ";
	}
	public String eat(){
		return hamburger.eat() + drip();
	}
}
```

```
// Dine.java  (the driver class that demonstrates the signatures)
public class Dine {
   public static void main(String args[]){
      Hamburger hamburger = new Hamburger();


      System.out.println("You ordered a " + hamburger.description() + ".");
      System.out.println("Thoughts: " + hamburger.eat() + "\n");
```

```
You ordered a hamburger.
Thoughts: Tasty!
```

```
// Dine.java  (the driver class that demonstrates the signatures)
public class Dine {
   public static void main(String args[]){
      Hamburger hamburger = new Hamburger();

      System.out.println("You ordered a " + hamburger.description() + ".");
      System.out.println("Thoughts: " + hamburger.eat() + "\n");

      hamburger = new Bacon(hamburger);
      hamburger = new Lettuce(hamburger);
      hamburger = new Tomato(hamburger);

      System.out.println("You ordered a " + hamburger.description() + ".");
      System.out.println("Thoughts: " + hamburger.eat() + "\n");
```

```
You ordered a hamburger.
Thoughts: Tasty!

You ordered a hamburger with bacon with lettuce with tomato.
Thoughts: Tasty! Sizzling! Crunchy! Juicy!
```

```
public class Dine {
    public static void main(String args[]){
        Hamburger hamburger = new Hamburger();

        System.out.println("You ordered a " + hamburger.description() + ".");
        System.out.println("Thoughts: " + hamburger.eat() + "\n");

        hamburger = new Bacon(hamburger);
        hamburger = new Lettuce(hamburger);
        hamburger = new Tomato(hamburger);

        System.out.println(hamburger.drip());   // causes an error

    }

}
```

```
public class Dine {

   public static void main(String args[]){

      Hamburger hamburger = new Hamburger();

          .

          .

          .

      hamburger = new EColi(hamburger);


      System.out.println("You ordered a " + hamburger.description() + ".");

      System.out.println("Thoughts: " + hamburger.eat() + "\n");

   }

}
```

```
You ordered a hamburger .

Thoughts: Tasty!


You ordered a hamburger  with bacon with lettuce with tomato.

Thoughts: Tasty! Sizzling! Crunchy! Juicy!


You ordered a hamburger  with bacon with lettuce with tomato with E. Coli.

Thoughts: E. Coli? Ha very funny. Gimme that. Tasty! Sizzling! Crunchy! Juicy!

One hour later: I don't feel so good...

Two hours later: Did I just poop my pants?
```

```
public class EColi extends HamburgerDecorator
{
	Hamburger hamburger;

	public EColi(Hamburger h){
		hamburger = h;
	}
	public String description(){
		return hamburger.description() + " with E. Coli";
	}
	public String causeNausea(){
		return "\nOne hour later: I don't feel so good...\n";
	}
	public String causeDiarrhea(){
		return "Two hours later: Did I just poop my pants? ";
	}
	public String eat(){
		return "E. Coli? Ha very funny. Gimme that. " + hamburger.eat()
		+ causeNausea() + causeDiarrhea();
	}
}
```