

CS101

Problem Solving and Object-Oriented Programming

L17: public/private, More Classes



Creating our own classes

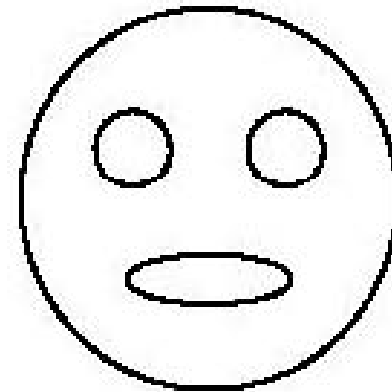
Consider making a funny face:

- Physical Characteristics:

- Head
- Mouth
- Two Eyes

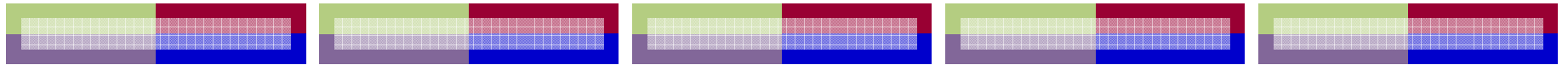
- Behaviors

- Can check contains
- Movable (by dragging)



- A FunnyFace class allows us to create new objects and lets us extend WindowController in a straight-forward way.





public vs private

- public – item can be used in classes other than the one in which it is declared.
- private – item can only be used within the class in which it is declared






public vs private

- public - can be used in classes other than the one in which it is declared.

```
public class Train {  
    public void move (double dx) {  
        ...  
    }  
}
```

```
public class TrainController {  
    private Train train;  
    ...  
    public void onMouseDrag (Location point) {  
        ...  
        train.move (distance);  
    }  
}
```






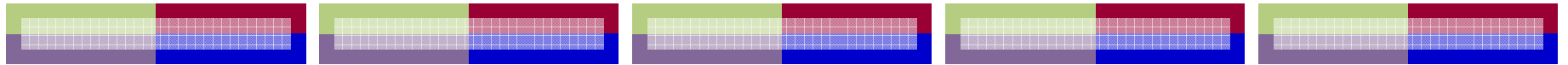
public vs private

- private - can only be used within the class in which it is declared

```
public class Train {  
    public void move (double dx) {  
        ...  
    }  
}
```

```
public class TrainController {  
    private Train train;  
    ...  
    public void onMouseDrag (Location point) {  
        ...  
        train.move (distance);  
    }  
}
```



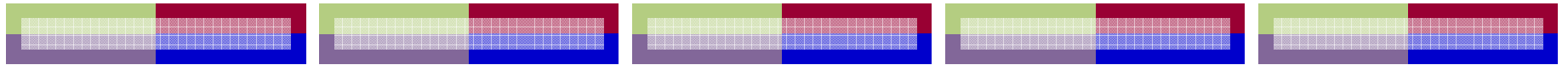


Producing Smoke

- How should the train produce a puff of smoke?
- What should the smoke look like?
- Where should it appear?

```
public void produceSmoke(...) {  
    ...  
}
```



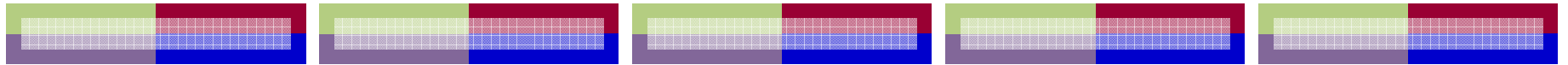


Producing Smoke

```
public class Train {  
    ...  
    private FramedRect smokeStack;  
  
    public Train (... , DrawingCanvas trainCanvas) {  
        ...  
        smokeStack = new FramedRect (... , trainCanvas);  
    }  
  
    public void produceSmoke () {  
        FilledOval smoke = new FilledOval (  
            smokestack.getX() + (SMOKESTACK_WIDTH - SMOKE_WIDTH) / 2,  
                smokestack.getY() - 1.5 * SMOKE_HEIGHT,  
                SMOKE_WIDTH, SMOKE_HEIGHT, trainCanvas);  
        smoke.setColor(SMOKE_COLOR);  
    }  
}
```

Where are the variables used in produceSmoke declared?

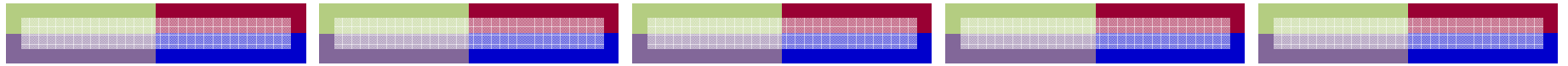




Producing Smoke

```
public class Train {  
    ...  
    private FramedRect smokeStack;  
    public Train (... , DrawingCanvas trainCanvas) {  
        ...  
        smokeStack = new FramedRect (... , trainCanvas);  
    }  
    public void produceSmoke () {  
        FilledOval smoke = new FilledOval (  
            smokeStack.getX() + (SMOKESTACK_WIDTH - SMOKE_WIDTH) / 2,  
            smokeStack.getY() - 1.5 * SMOKE_HEIGHT,  
            SMOKE_WIDTH, SMOKE_HEIGHT, trainCanvas);  
        smoke.setColor(SMOKE_COLOR);  
    }  
}
```

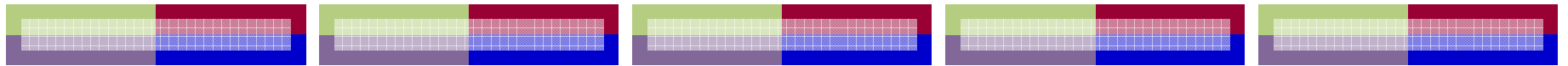




Producing Smoke

```
public class Train {  
    ...  
    private FramedRect smokeStack;  
  
    public Train (... , DrawingCanvas trainCanvas) {  
        ...  
        smokeStack = new FramedRect (... , trainCanvas);  
    }  
  
    public void produceSmoke () {  
        FilledOval smoke = new FilledOval (  
            smokestack.getX() + (SMOKESTACK_WIDTH - SMOKE_WIDTH) / 2,  
            smokestack.getY() - 1.5 * SMOKE_HEIGHT,  
            SMOKE_WIDTH, SMOKE_HEIGHT, trainCanvas);  
        smoke.setColor(SMOKE_COLOR);  
    }  
}
```

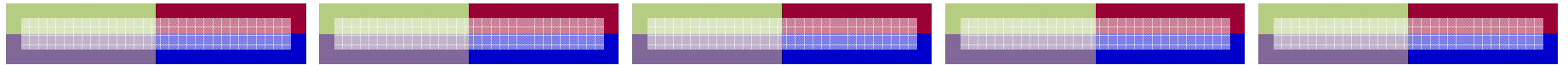




Producing Smoke

```
public class Train {  
    ...  
    private FramedRect smokeStack;  
  
    public Train (... , DrawingCanvas trainCanvas) {  
        ...  
        smokeStack = new FramedRect (... , trainCanvas);  
    }  
  
    public void produceSmoke () {  
        FilledOval smoke = new FilledOval (  
            smokestack.getX() + (SMOKESTACK_WIDTH - SMOKE_WIDTH) / 2,  
            smokestack.getY() - 1.5 * SMOKE_HEIGHT,  
            SMOKE_WIDTH, SMOKE_HEIGHT, trainCanvas);    ???  
        smoke.setColor(SMOKE_COLOR);  
    }  
}
```





Purpose of Constructors

- Put something on the display (e.g. by creating FilledOvals)
- Remembering things created for later use (like the canvas so that we can create new shapes in other methods)

```
private DrawingCanvas smokeCanvas;
```


```
public Train (double left, double trackHeight, DrawingCanvas trainCanvas) {  
    car = new FilledRect (left, top, CAR_WIDTH, CAR_HEIGHT, trainCanvas);  
    ...  
    smokeCanvas = trainCanvas;  
}
```

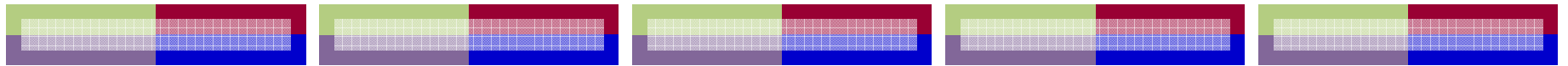
```
public void produceSmoke () {  
    FilledOval smoke = new FilledOval (... , smokeCanvas);  
    ...  
}
```





Midterm: Thursday April 9

- In class, closed book, closed notes, but I'll provide a cheat sheet
 - Tuesday, April 7, Review in class - come with questions!
 - Covers only Java, material through Scope (March 31 + public/private of today)
 - Sample types of questions:
 - What does it do? - like the mystery programs
 - Vocabulary - I provide a program and ask you to identify all parameters, all methods, etc.
 - Find a bug - I provide a program and ask you to find the bug in it.
 - Declare variables - I provide a program and ask you to insert appropriate variable declarations
 - Write a short method
- 

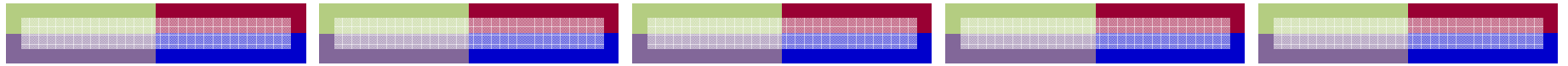


Variable Initialization

- Variables must be assigned to before they can be used.

```
public Train (double left, double trackHeight, DrawingCanvas canvas) {  
    car = new FilledRect(left, trackHeight - CAR_HEIGHT -  
        WHEEL_SIZE, CAR_WIDTH, CAR_HEIGHT, canvas);  
    ...  
    smokestack = new FilledRect (left + CAR_WIDTH - 2 *  
        SMOKESTACK_WIDTH, car.getY() - SMOKESTACK_HEIGHT,  
        SMOKESTACK_WIDTH, SMOKESTACK_HEIGHT, canvas);  
    smoke = new FilledOval (smokestack.getX() +  
        (SMOKESTACK_WIDTH - SMOKE_WIDTH) / 2,  
        smokestack.getY() - 1.5 * SMOKE_HEIGHT, SMOKE_WIDTH,  
        SMOKE_HEIGHT, canvas);  
    ...  
}
```





NullPointerException

- Occurs if variable is used before it is assigned to.

```
public Train (double left, double trackHeight, DrawingCanvas canvas) {  
    smokestack = new FilledRect (left + CAR_WIDTH - 2 *  
        SMOKESTACK_WIDTH, car.getY() - SMOKESTACK_HEIGHT,  
        SMOKESTACK_WIDTH, SMOKESTACK_HEIGHT, canvas);  
    car = new FilledRect(left, trackHeight - CAR_HEIGHT -  
        WHEEL_SIZE, CAR_WIDTH, CAR_HEIGHT, canvas);  
    ...  
}
```

Exception in thread "AWT-EventQueue-0" java.lang.NullPointerException
at Train.<init>(Train.java:46)






Constructors & Instance Variables

- Instance variable initializations cannot depend on statements in the constructor

```
private smokestackTop = car.getY() - SMOKESTACK_HEIGHT;
```

```
public Train (double left, double trackHeight, DrawingCanvas canvas) {  
    smokestack = new FilledRect (left + CAR_WIDTH - 2 *  
        SMOKESTACK_WIDTH, smokestackTop,  
        SMOKESTACK_WIDTH, SMOKESTACK_HEIGHT, canvas);  
    car = new FilledRect(left, trackHeight - CAR_HEIGHT -  
        WHEEL_SIZE, CAR_WIDTH, CAR_HEIGHT, canvas);  
    ...  
}
```

Exception in thread "AWT-EventQueue-0" java.lang.NullPointerException
at Train.<init>(Train.java:41)





Declaring Duplicate Variables


- Be careful not to re-declare a variable

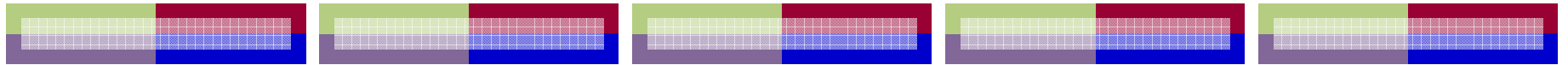
```
private FilledRect car;
```

```
public Train (double left, double trackHeight, DrawingCanvas canvas) {  
    FilledRect car = new FilledRect(left, trackHeight - CAR_HEIGHT -  
        WHEEL_SIZE, CAR_WIDTH, CAR_HEIGHT, canvas);  
    ...  
}
```

```
public boolean contains (Location point) {  
    if (car.contains (point)) {  
        return true;  
    }  
}
```

Exception in thread "AWT-EventQueue-0" java.lang.NullPointerException
at Train.contains(Train.java:66)





Null

- A special value that a variable / parameter can have that means “no value”
- Not applicable to int, double or boolean
- Sending a message to a variable when its value is null results in `NullPointerException`!
- Can check for null using an if-statement:
 - ```
if (train != null) {
 train.move(...);
 ...
}
```

