Programming!

Connecting hardware with software

Ashley David, FRC Team 503
August 26, 2017

What is Java and how it's used

- One of the many programming languages
  - Python
  - Swift
  - C++
- Everywhere!!
  - Usernames and passwords
  - Barcodes
  - Apps

How does a robotics game work?

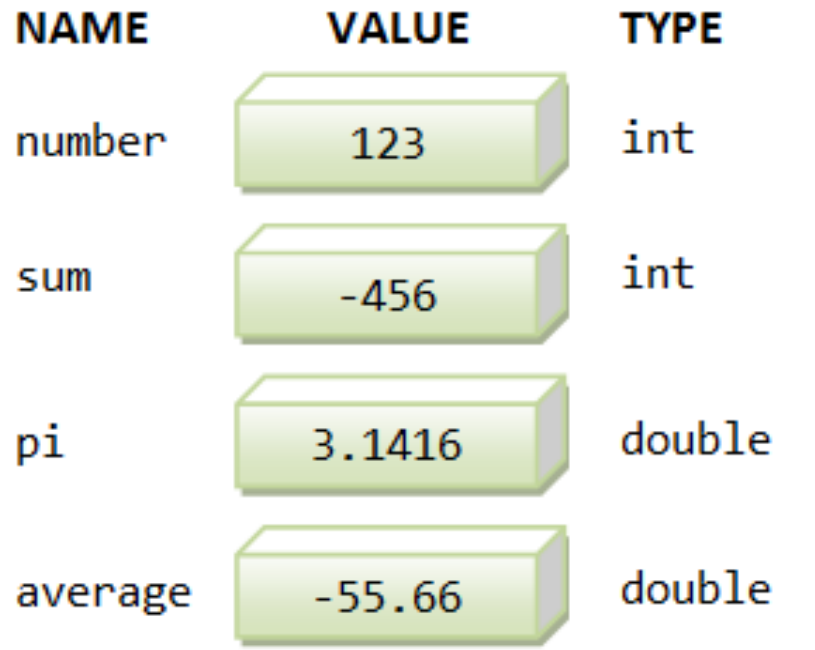
- First 30 seconds is called auton, where the robot moves without help
- Next 60 seconds is called teleop where the driver and operator drive the robot and try to get points
- Last 30 seconds is called endgame where the driver tries to gain points any way they can

Teleop!

Variables

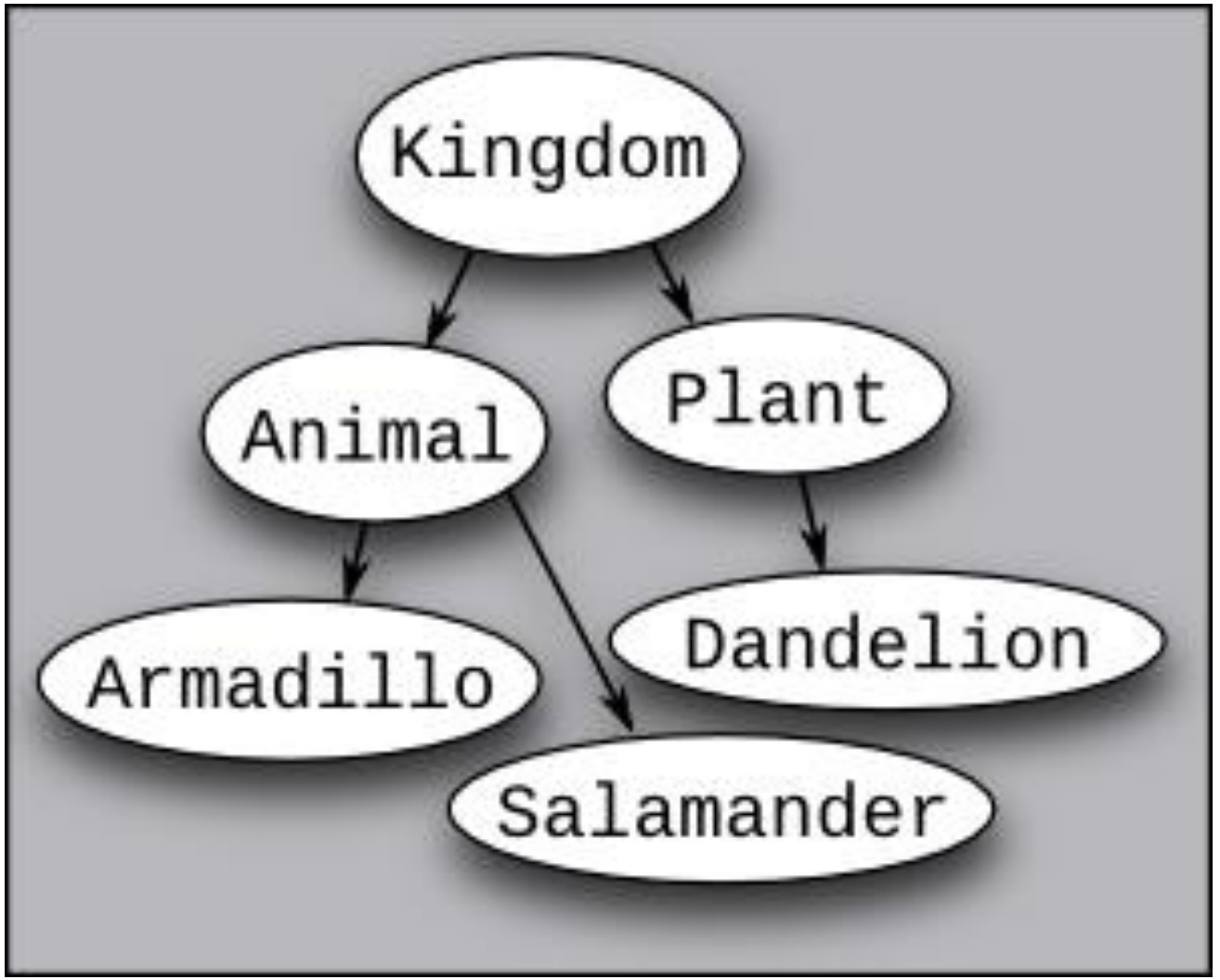
- A number that has at least 1 number after the decimal; has to be initialized at the beginning of the code or it will cause an error
- Example:

```
double left = 0;
double right = 1;
```



A variable has a *name*, stores a *value* of the declared *type*

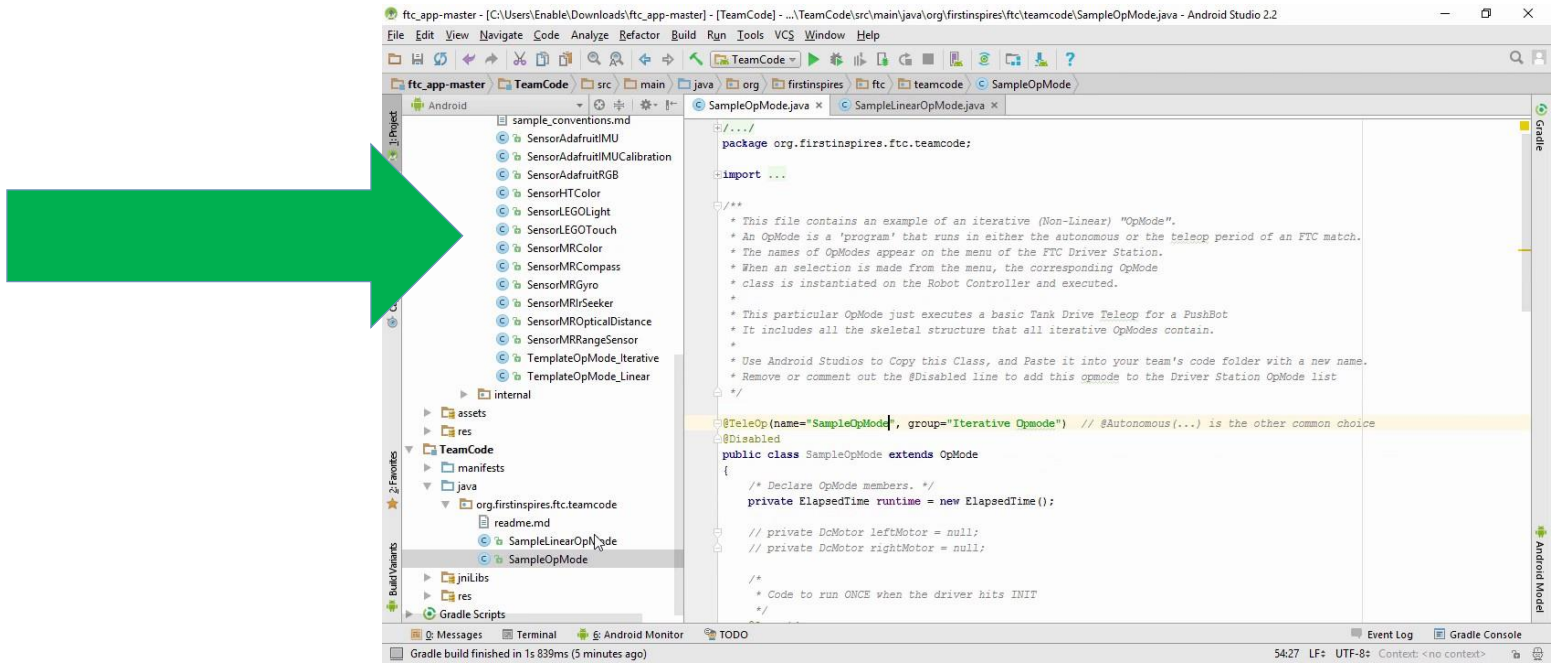
Class Hierarchy



OpMode

- A class located within the FTC SDK (robot controller app source code)
- Example:

```
public class GWCTeleopTest extends OpMode{
```



FROG FORCE 503 A LEAP AHEAD

@Teleop

- Explains to the phone you're running the code on, if the code should be run during Auton or Teleop (what is actually seen on the phone)

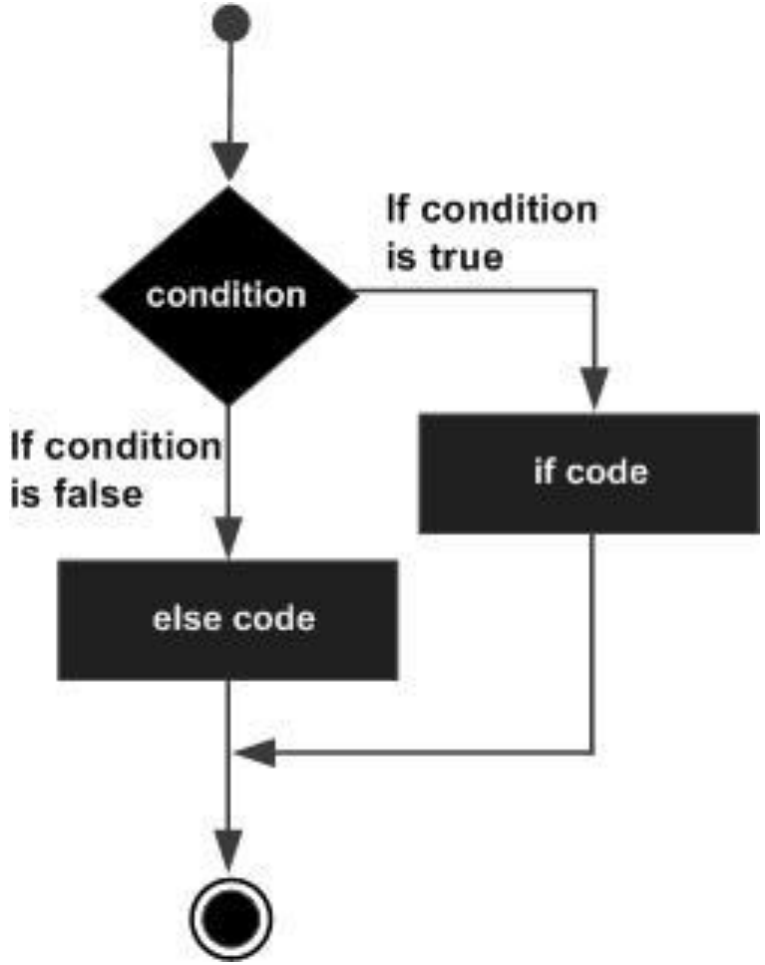
```
@Teleop (name=Pushbot: GWC Teleop Test", group="Pushbot")
```

FROG FORCE 503 A LEAP AHEAD

if / if else

- It tells your program to execute a certain section of code only if a particular test evaluates to true.
- Example :

```
if (gamepad2.y)
      robot.armMotor.setPower(robot.ARM_UP_POWER);
else if (gampad2.a)
      robot.armMotor.setPower(robot.ARM_DOWN_POWER);
else
      robot.armMotor.setPower(0.0);
```



Telemetry

- A way to output the distance the robot is moving without actual coordinates

- Example:

```
telemetry.addData("left", left);
```

init()

- It is used to declare/initialize the common parts of a class.

- Example:

```
public void init() {
        */ Initialize the hardware variables
                }
```

init_loop()

- init_loop initializes the loop
  - A Loop: Repeats a statement or group of statements while a given condition is true

- Example :

```
public void init_loop()
```

@Override

- The method is overriding the parent class

- Example:

```
@Override
```

setPower

- It’s one of the most common ways to run a motor
- Example:

```
double left;
left = -gamepad1.left_stick_y;
robot.leftMotor.setPower(left);
```

@Disabled

- When you don’t want your program to be seen on the phone you disable it, but if you want it to be seen, you “un-disable” the program, therefore enabling it
- Example :

```
//@Disabled (to see the program)
@Disabled (to not see the program)
```

Auton!

new

- Creates a new instance of the class

- Example:

```
timer = new ElapsedTime ()
```

void

- The method has no return value
  - The result of a function that returns normally, but does not provide a result value to its caller

- Example :

```
public void loop()
```

timer

- Counts how many seconds the robot does a specific function

- Example:

```
timer = new ElapsedTime();
```

.reset

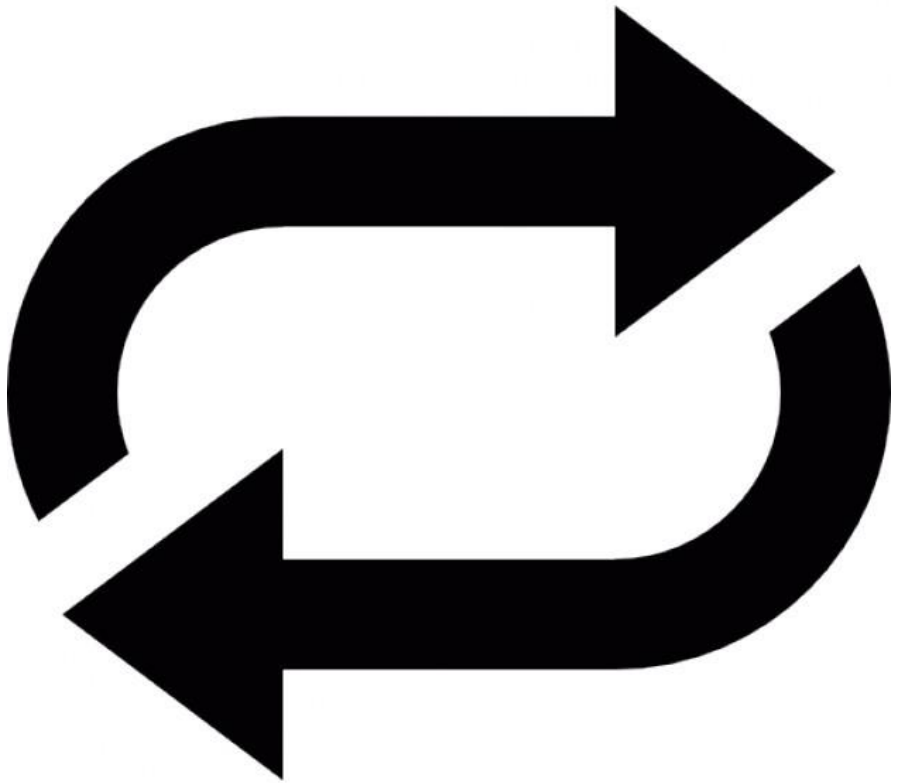
- The method that resets the timer

- Example:

```
timer.reset();
```

loop

- Repeats a statement or group of statements while a given condition is true



- Example:

```
public void init_loop()
```

How to Move Forward & Turn

```
public void loop() {
  if (masterState == 0){
      if (timer.seconds() <= 5.0){
          robot.leftMotor.setPower(1.0);
          robot.rightMotor.setPower(1.0)};
      else {
          masterState = 1};
  }
  else if (masterState == 1){
      if (timer.seconds() <= 5.66){
          robot.leftMotor.setPower(1.0);
          robot.rightMotor.setPower(-1.0);
      }
      else {
          masterState = 2;
      }
  }
else if(masterState == 2) {
      if (timer.seconds() <= 8.0){
          robot.leftMotor.setPower(1.0);
          robot.rightMotor.setPower(1.0);
      }
      else {
          robot.leftMotor.setPower(0.0);
          robot.rightMotor.setPower(0.0);

          masterState = 3;
      }
  }
telemetry.addData("state",masterState);
  }
```

Questions?

What we are doing today will transform tomorrow's culture.