Computer Programming Fundamentals

CS 152
Professor: Leah Buechley
TAs: Melody Horn, Noah Garcia, Andrew Geyko, Juan Ormaza
Time: MWF 10:00-10:50am
https://handandmachine.cs.unm.edu/classes/CS152_Fall2021/

ASSIGNMENT 5

Due Monday 11/8

Start early!

questions?

OPEN IntelliJ

CREATE A NEW PROJECT
“Week11”

REINSTALL TURTLE LIBRARY

CREATE A “BasicPanel.java” CLASS

COPY AND PASTE BasicPanel.java FROM CLASS SCHEDULE

L-SYSTEM IN CODE

CREATE AN L-SYSTEM CLASS

L-Systems

A set of rules that turn a simple starting word into a more complex word by replacing symbols with more complex patterns.

Example:

starting word: F
rule: F-F++F-F

F
F-F++F-F
F-F++F-F-F-F++F-F++F-F++F-F-F-F++F-F

LSystem.java

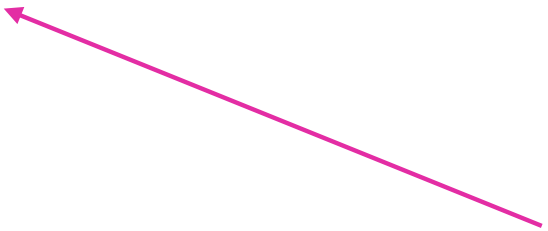
```
public class LSystem {
    String startingWord, computedWord;
    String[][] rules = {{"F", “F-F++F-F"}};
}
```

will hold computed “words”

a list of rules
each rule is a 2 item array
only one rule for now:
F —> F-F++F-F

LSystem.java

```
public class LSystem {
    String startingWord, computedWord;
    String[][] rules = {{"F", “F-F++F-F"}};
    int iterations;
}
```



number of iterations executed

questions?

LSystem.java

```
public class LSystem {
    String startingWord, computedWord;
    String[][] rules = {{"F", "F-F++F-F"}};
    int iterations;

    LSystem() {
        startingWord = "F";
        computedWord = startingWord;
        iterations = 0;
    }
}
```

NOW WE NEED TO DO COMPUTATION

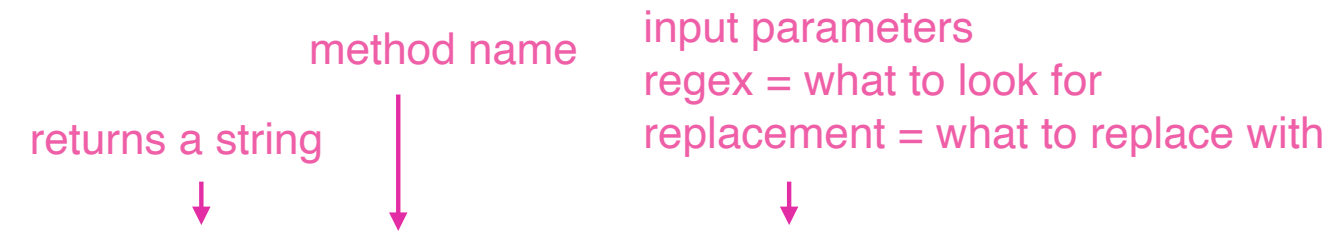
REPLACE F with F-F++F-F

JAVA STRING METHODS

https://www.w3schools.com/java/java_ref_string.asp

https://docs.oracle.com/en/java/javase/16/docs/api/java.base/java/lang/String.html

JAVA replaceAll METHOD




```
public String replaceAll(String regex, String replacement)
```

Replaces each substring of this string that matches "regex" with the given "replacement".

Parameters:
`regex` - the regular expression to which this string is to be matched
`replacement` - the string to be substituted for each match

Returns:
The resulting `String`

https://docs.oracle.com/en/java/javase/16/docs/api/java.base/java/lang/String.html#replaceAll(java.lang.String,java.lang.String)

TESTING OUT replaceAll()

```
public class LSystem {
    String startingWord, computedWord;
    String[][] rules = {{"F", "F-F++F-F"}};
    int iterations;

    LSystem() {
        startingWord = "F";
        computedWord = startingWord;
        iterations = 0;
    }

    public static void main(String[] args) {
        String a = "hello there!";
        String b = a.replaceAll("ll", "llllllll");
        System.out.println(a);
        System.out.println(b);
    }


}
```

```
hello there!
hellllllllo there!
```

questions?

USING replaceAll to REPLACE F with F-F++F-F

LSystem.java

```
public class LSystem {
    String startingWord, computedWord;
    String[][] rules = {{"F", "F-F++F-F"}};
    int iterations;

    LSystem() {
        startingWord = "F";
        computedWord = startingWord;
        iterations = 0;
    }

   void iterate() {
        //replace every "F" in word with "F-F++F-F"
    }

}
```

LSystem.java

```
public class LSystem {
    String startingWord, computedWord;
    String[][] rules = {{"F", "F-F++F-F"}};
    int iterations;

    LSystem() {
        startingWord = "F";
        computedWord = startingWord;
        iterations = 0;
    }

   void iterate() {
        String temporaryWord = computedWord;

    }

}
```

create a variable to
store the current word in

LSystem.java

```
public class LSystem {
    String startingWord, computedWord;
    String[][] rules = {{"F", "F-F++F-F"}};
    int iterations;

    LSystem() {
        startingWord = "F";
        computedWord = startingWord;
        iterations = 0;
    }

   void iterate() {
        String temporaryWord = computedWord;
        computedWord = temporaryWord.replaceAll(rules[0][0], rules[0][1]);
    }

}
```

do the replacement
storing the new word in
the computedWord variable

LSystem.java

```
public class LSystem {
    String startingWord, computedWord;
    String[][] rules = {{"F", "F-F++F-F"}};
    int iterations;

    LSystem() {
        startingWord = "F";
        computedWord = startingWord;
        iterations = 0;
    }

   void iterate() {
        String temporaryWord = computedWord;
        computedWord = temporaryWord.replaceAll(rules[0][0], rules[0][1]);
        iterations++;
        System.out.println("interation " + (iterations) + ": " + computedWord);
    }

}
```

increment the iterations
variable and
print out the new word

LSystem.java

```
public class LSystem {
    String startingWord, computedWord;
    String[][] rules = {{"F", "F-F++F-F"}};
    int iterations;

    LSystem() {
        startingWord = "F";
        computedWord = startingWord;
        iterations = 0;
        System.out.println("interation " + (iterations) + ": " + computedWord);
    }

   void iterate() {
        String temporaryWord = computedWord;
        computedWord = temporaryWord.replaceAll(rules[0][0], rules[0][1]);
        iterations++;
        System.out.println("interation " + (iterations) + ": " + computedWord);
    }

}
```

add a print statement to constructor to print starting word

questions?

LET’S TRY IT OUT

IN main

```
public static void main(String[] args) {
    LSystem l = new LSystem();
    l.iterate();
}
```

```
interation 0: F
interation 1: F-F++F-F
```

IN main

```
public static void main(String[] args) {
    LSystem l = new LSystem();
    for (int i=0;i<3;i++) {
        l.iterate();
    }
}
```

```
interation 0: F
interation 1: F-F++F-F
interation 2: F-F++F-F-F-F++F-F++F-F++F-F-F-F++F-F
interation 3: F-F++F-F-F-F++F-F++F-F++F-F-F-F++F-F-F-F++F-F-F-F++F-F++F-F++F-F-F-F++F-F-F-F++F-F++F-F++F-F-F-F++F-F-F-F++F-F-F-F++F-F-F-F++F-F++F-F++F-F-F-F++F-F-F-F++F-F-F-F++F-F-F-F++F-F++F-F++F-F-F-F++F-F
```

questions?

VISUALIZING THE L-System

L-Systems and Turtle Geometry

Visualize the pattern by translating it into actions carried out by the turtle. Each symbol is translated into an action

Example:

F = move forward an amount (100)
- = turn left an angle (60°)
+ = turn right an angle (60°)

F
F-F++F-F
F-F++F-F-F-F++F-F++F-F++F-F-F-F++F-F

A draw METHOD

A draw METHOD

```
void draw(Turtle t) {

}
```

takes a Turtle object as an input
this turtle will do the drawing

A draw METHOD

```
void draw(Turtle t, double size, double angle) {

}
```

size and angle inputs
other drawing parameters

WE WANT TO TRANSLATE EACH CHARACTER IN FINAL WORD INTO A MOVEMENT

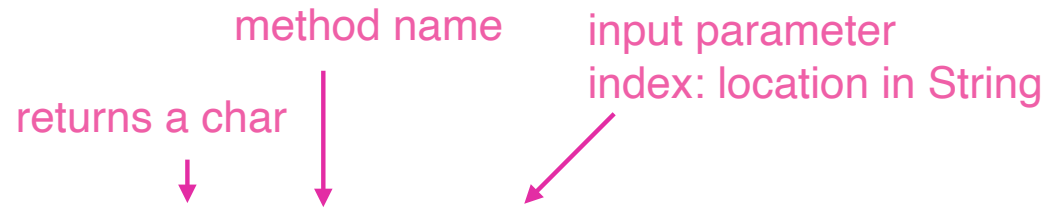
HOW??

JAVA STRING METHODS

https://www.w3schools.com/java/java_ref_string.asp

https://docs.oracle.com/en/java/javase/16/docs/api/java.base/java/lang/String.html

JAVA charAt METHOD




```
public char charAt(int index)
```

Returns the `char` value at the specified index. An index ranges from `0` to `length() - 1`. The first `char` value of the sequence is at index `0`, the next at index `1`, and so on, as for array indexing.

Parameters:
`index` - the index of the `char` value.

Returns:
the `char` value at the specified index of this string. The first `char` value is at index `0`.

https://docs.oracle.com/en/java/javase/16/docs/api/java.base/java/lang/String.html#charAt(int)

TESTING OUT charAt()

```
public static void main(String[] args) {
    //LSystem l = new LSystem();
    //for (int i=0;i<3;i++) {
    //    l.iterate();
    //}

    String a = "hello";
    char b = a.charAt(0);
    System.out.println("charAt 0: " +b);
}
```

```
charAt 0: h
```

TESTING OUT charAt()

```
public static void main(String[] args) {
    //LSystem l = new LSystem();
    //for (int i=0;i<3;i++) {
    //    l.iterate();
    //}

    String a = "hello";
    char b = a.charAt(1);
    System.out.println("charAt 1: " +b);
}
```

```
charAt 1: e
```

questions?

USING charAt to TRANSLATE EACH CHARACTER INTO A MOVEMENT

A draw METHOD

```
void draw(Turtle t, double size, double angle) {
    for (int i = 0; i < computedWord.length(); i++) {

    }
}
```

loop through entire word

A draw METHOD

```
void draw(Turtle t) {
    for (int i = 0; i < computedWord.length(); i++) {
        if (computedWord.charAt(i) == 'F')
   }
}
```

see if the character at the current position is an ‘F’

A draw METHOD

```
void draw(Turtle t, double size, double angle) {
    for (int i = 0; i < computedWord.length(); i++) {
        if (computedWord.charAt(i) == 'F')
            t.forward(size);
    }
}
```

if an ‘F’
move forward

A draw METHOD

```
void draw(Turtle t, double size, double angle) {
    for (int i = 0; i < computedWord.length(); i++) {
        if (computedWord.charAt(i) == 'F')
            t.forward(size);
        else if (computedWord.charAt(i) == '-')
            t.left(angle);
    }
}
```

if a '-'
turn left

A draw METHOD

```
void draw(Turtle t, double size, double angle) {
    for (int i = 0; i < computedWord.length(); i++) {
        if (computedWord.charAt(i) == 'F')
            t.forward(size);
        else if (computedWord.charAt(i) == '-')
            t.left(angle);
        else if (computedWord.charAt(i) == '+')
            t.right(angle);
    }
}
```

if a ‘+’
turn right

questions?

PUTTING IT ALL TOGETHER

CREATE AN “LSystemVis.java” CLASS

LSystemVis.java

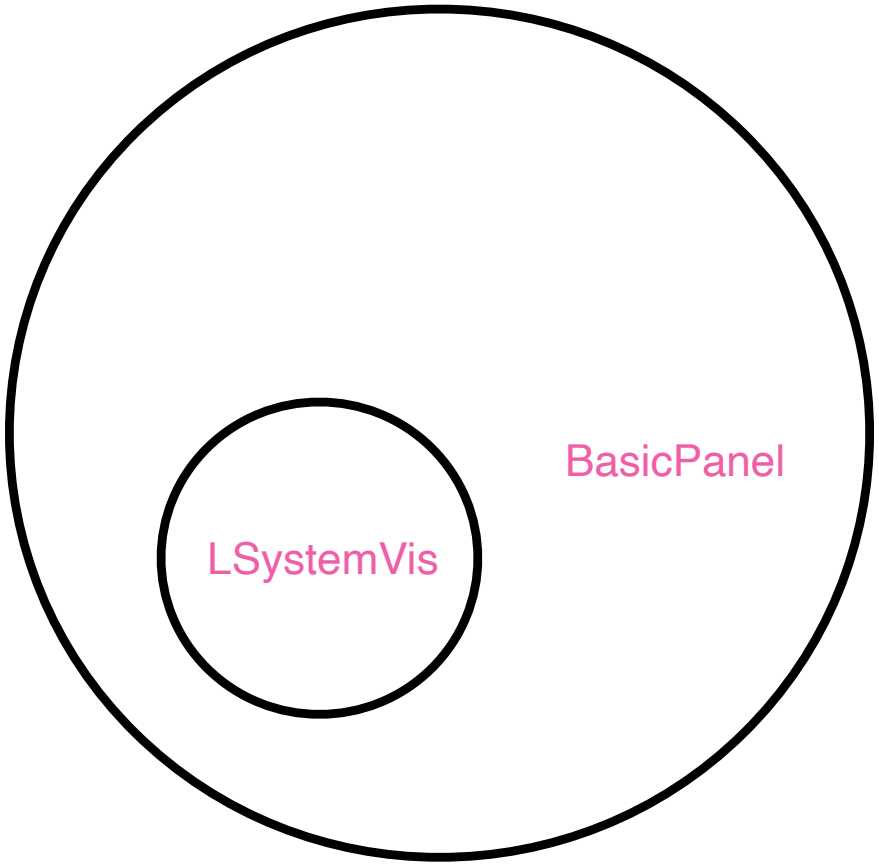
```
public class LSystemVis {

}
```

MAKE LSystemVis A SUBCLASS OF BasicPanel

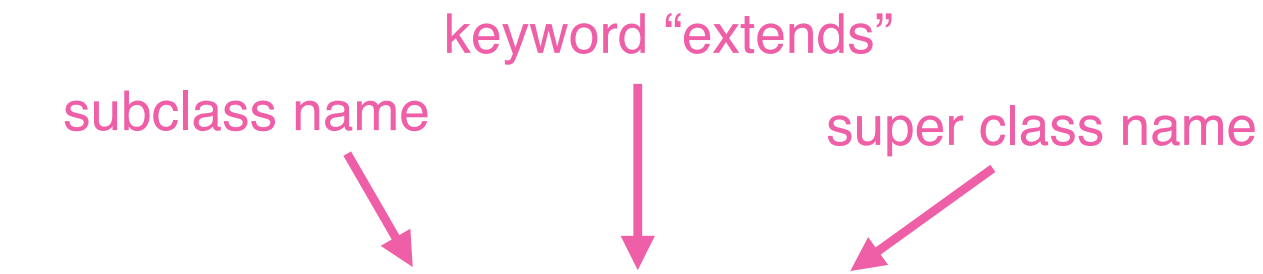
REVIEW

JAVA INHERITANCE



- LSystemVis is a “subclass” of BasicPanel &BasicPanel is a “super” class of LSystemVis
- All LSystemVis objects are BasicPanel objects
- LSystemVis objects contain and have access to all methods and variables defined in BasicPanel
- LSystemVis class can define additional variables and methods that are not part of BasicPanel

CREATING A SUBCLASS




```
public class LSystemVis extends BasicPanel {

}
```

LSystemVis.java

```
public class LSystemVis extends BasicPanel {

}
```

LSystemVis
will inherit all the functionality
that BasicPanel has

LSystemVis.java

```
public class LSystemVis extends BasicPanel {
    Turtle t;
    LSystem l;
}
```

add Turtle and LSystem variables

LSystemVis.java

```
public class LSystemVis extends BasicPanel {
    Turtle t;
    LSystem l;
    double size, angle;
}
```

add size and angle variables

LSystemVis.java

```
public class LSystemVis extends BasicPanel {
    Turtle t;
    LSystem l;
    double size, angle;

    LSystemVis() {
        t = new Turtle(this);
        l = new LSystem();
        size = 100;
        angle = 60;
    }
}
```

add a constructor

LSystemVis.java

```
public class LSystemVis extends BasicPanel {
    Turtle t;
    LSystem l;
    double size, angle;

    LSystemVis() {
        t = new Turtle(this);
        l = new LSystem();
        size = 100;
        angle = 60;
        l.draw(t,size,angle);
    }
}
```

draw iteration 0

LSystemVis.java

```
public class LSystemVis extends BasicPanel {
    Turtle t;
    LSystem l;
    double size, angle;

    LSystemVis() {
        t = new Turtle(this);
        l = new LSystem();
        size = 100;
        angle = 60;
        l.draw(t,size,angle);
    }

    @Override
    protected void paintComponent(Graphics g) {
        super.paintComponent(g);
        setBackground(Color.WHITE);
   }

}
```

add a paintComponent method

LSystemVis.java

```
public class LSystemVis extends BasicPanel {
    Turtle t;
    LSystem l;
    double size, angle;

    LSystemVis() {
        t = new Turtle(this);
        l = new LSystem();
        size = 100;
        angle = 60;
        l.draw(t,size,angle);
    }

    @Override
    protected void paintComponent(Graphics g) {
        super.paintComponent(g);
        setBackground(Color.WHITE);
        t.drawPath(g);
   }

}
```

draw the turtle's path

COMPILE AND RUN

DRAWS STARTING WORD/ITERATION 0

```
interation 0: F
```