Announcements

- Lab 3 due today/yesterday
- HW 3 due Mon.
- Lab 4 online soon. Leave time to read it carefully.
- MORE EMAIL! SEND EMAIL!
Today’s Plan

- Fun with Strings and while loops
String methods()

- `str.length() : int ≥ 0`
- `str.startsWith(otherString) : boolean`
- `str.equals(otherString) : boolean`
- `str.indexOf(otherString) : int ≥ -1`
- `str.indexOf(otherString, startIndex) : int ≥ -1`
- `str.contains(otherString) : boolean`
- `str.substring(startIndex, endIndex) : String`
- `str.substring(startIndex) : String`

https://docs.oracle.com/javase/7/docs/api/index.html?java/lang/String.html
Moving on...

- Initially all execution was linear
- When if-else statements, we introduced branching
- While loops also us to repeat statements
DEAR VARIOUS PARENTS, GRANDPARENTS, CO-WORKERS, AND OTHER "NOT COMPUTER PEOPLE."

WE DON'T MAGICALLY KNOW HOW TO DO EVERYTHING IN EVERY PROGRAM. WHEN WE HELP YOU, WE'RE USUALLY JUST DOING THIS:

---

START

FIND A MENU ITEM OR BUTTON WHICH LOOKS RELATED TO WHAT YOU WANT TO DO.

I CAN'T FIND ONE

PICK ONE AT RANDOM

I'VE TRIED THEM ALL.

OK

CLICK IT

HAVE YOU BEEN TRYING THIS FOR OVER HALF AN HOUR?

NO

NO

ASK SOMEONE FOR HELP OR GIVE UP.

YES

DID IT WORK?

NO

YOU'RE DONE!

YES

YOU'RE DONE!

GOOGLE THE NAME OF THE PROGRAM PLUS A FEW WORDS RELATED TO WHAT YOU WANT TO DO. FOLLOW ANY INSTRUCTIONS.

---

PLEASE PRINT THIS FLOWCHART OUT AND TAPE IT NEAR YOUR SCREEN. CONGRATULATIONS; YOU'RE NOW THE LOCAL COMPUTER EXPERT!

Courtesy xkcd.com
line = toServer.in.nextLine();

while ( ! line.equals(".") ) {
    message.append( line + "\n" );
    line = toServer.in.nextLine();
}
If Statements

if ( condition ) {
    statements to be executed if condition is true
} else {
    statements to be executed if condition is false
}
If Statements

if ( condition )
  statement to be executed if condition is true
else
  statement to be executed if condition is false
While Loops

while ( condition ) {
    statements to be repeated
}

While Loops

while ( condition )
statement to be repeated
While Loops

statement(s) to initialize loop variable(s)
while ( condition ) {
    statement(s) to perform one step
    statement(s) to move onto next step
}
statement(s) to wrap things up nicely
line = toServer.in.nextLine();

while (!line.equals(".")) {
    message.append(line + "\n");
    line = toServer.in.nextLine();
}
while ( condition ) {
    statement(s) to be perform one step
    statement(s) to move onto next step
}

statement(s) to wrap things up nicely

line = toServer.in.nextLine();

while ( ! line.equals(".")) {
    message.append( line + "\n" );
    line = toServer.in.nextLine();
}

statement(s) to initialize loop variable(s)
while ( condition ) {
    statement(s) to be perform one step
    statement(s) to move onto next step
}
statement(s) to wrap things up nicely

line = toServer.in.nextLine();

while ( ! line.equals(".")) {
    message.append( line + "\n" );
    line = toServer.in.nextLine();
}
statement(s) to initialize loop variable(s)

while ( condition ) {
    statement(s) to be perform one step
    statement(s) to move onto next step
}

statement(s) to wrap things up nicely

line = toServer.in.nextLine();

while ( ! line.equals(".")) {
    message.append( line + "\n" );
    line = toServer.in.nextLine();
}
statement(s) to initialize loop variable(s)
while ( condition ) {
  statement(s) to be perform one step
  statement(s) to move onto next step
}
statement(s) to wrap things up nicely

  line = toServer.in.nextLine();

  while ( ! line.equals(".")) {

    message.append( line + "\n" );

    line = toServer.in.nextLine();
  }
statement(s) to initialize loop variable(s)
while ( condition ) {
    statement(s) to be perform one step
    statement(s) to move onto next step
}
statement(s) to wrap things up nicely

line = toServer.in.nextLine();

while ( ! line.equals(".")) {
    message.append( line + "\n" );
    line = toServer.in.nextLine();
}
statement(s) to initialize loop variable(s)
while (condition) {
    statement(s) to be perform one step
    statement(s) to move onto next step
}
statement(s) to wrap things up nicely

line = toServer.in.nextLine();

while (!line.equals(".")) {
    message.append(line + "\n");
    line = toServer.in.nextLine();
}
statement(s) to initialize loop variable(s)
while ( condition ) {
    statement(s) to be perform one step
    statement(s) to move onto next step
}
statement(s) to wrap things up nicely
statement(s) to initialize loop variable(s)
while ( condition ) {
    statement(s) to be perform one step
    statement(s) to move onto next step
}
statement(s) to wrap things up nicely

int positionOfSpace = words.indexOf( " " );
while ( positionOfSpace >= 0 ) {
    String word = words.substring(0, positionOfSpace );
    wordMenu.add(
wordMenu.addItem( word );

    words = words.substring( positionOfSpace + 1 ).trim();
    positionOfSpace = words.indexOf( " " );
}
wordMenu.addItem( words );
statement(s) to initialize loop variable(s)
while ( condition ) {
    statement(s) to be perform one step
    statement(s) to move onto next step
}
statement(s) to wrap things up nicely

int positionOfSpace = words.indexOf( " " );
while ( positionOfSpace >= 0 ) {
    String word = words.substring(0, positionOfSpace);
    wordMenu.addItem( word );
    words = words.substring( positionOfSpace + 1 ).trim();
    positionOfSpace = words.indexOf( " " );
}
wordMenu.addItem( words );
int positionOfSpace = words.indexOf( " " );
while ( positionOfSpace >= 0 ) {
    String word = words.substring(0, positionOfSpace );
    wordMenu.addItem( word );
    words = words.substring( positionOfSpace + 1 ).trim();
    positionOfSpace = words.indexOf( " " );
}
wordMenu.addItem( words );
```java
int positionOfSpace = words.indexOf(" ");
while ( positionOfSpace >= 0 ) {
    String word = words.substring(0, positionOfSpace);
    wordMenu.addItem(word);
    words = words.substring(positionOfSpace + 1).trim();
    positionOfSpace = words.indexOf(" ");
}
wordMenu.addItem(words);
```
statement(s) to initialize loop variable(s)
while ( condition ) {
  statement(s) to be perform one step
  statement(s) to move onto next step
}
statement(s) to wrap things up nicely

int positionOfSpace = words.indexOf(" ");
while ( positionOfSpace >= 0 ) {
  String word = words.substring(0, positionOfSpace);
  wordMenu.addItem(word);
  words = words.substring(positionOfSpace + 1).trim();
}
wordMenu.addItem(words);
int positionOfSpace = words.indexOf( " " );
while ( positionOfSpace >= 0 ) {
    String word = words.substring(0, positionOfSpace);
    wordMenu.addItem( word );
    words = words.substring( positionOfSpace + 1 ).trim();
    positionOfSpace = words.indexOf( " " );
}
wordMenu.addItem( words );
Gray Matters
statement(s) to initialize loop variable(s)
while ( condition ) {
    statement(s) to be perform one step
    statement(s) to move onto next step
}
statement(s) to wrap things up nicely
int brightness = 0;
while ( brightness <= BRIGHTEST_SHADE ) {

    field = new JTextField("" + brightness, WIDTH );
    contentPane.add( field );

    field.setBackground(
        new Color( brightness, brightness, brightness, brightness )
    );

    brightness = brightness + BRIGHTEST_SHADE/51;
}