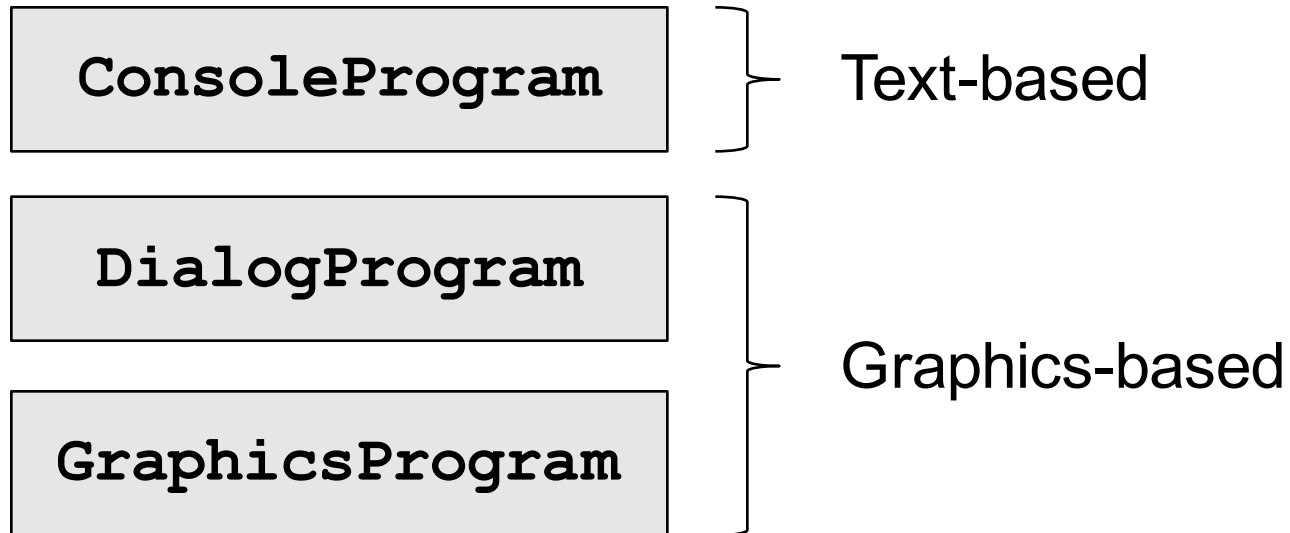


# Programming – Lecture 15

Going Beyond  
the ACM Library

# The ACM Library

So far, we have used the ACM library to write our programs. Most of our programs were instances of one of these classes:



# Alternatives

**ConsoleProgram**

**DialogProgram**

**GraphicsProgram**

# Alternatives: Text-Based

Simple Java programs are text-based by default. Remember lecture 2:

```
public class HelloWorld {  
    public static void main (String[] args) {  
        System.out.println("Hello World!");  
    }  
}
```

# Class Definition

- ACM:

```
public class Wubbel
    extends ConsoleProgram {
    //...
}
```

- Java:

No restrictions. Every class can have a **main** method.

# Program Entry Point

- ACM:

```
public void run() {  
    //...  
}
```

- Java:

```
public static void main(String[] args) {  
    //...  
}
```

# Console Output

- ACM:

```
println(...);
```

- Java:

```
System.out.println(...);
```

# Console Input

- ACM:

```
String s = readLine();  
int i = readInt();
```

- Java:

```
Scanner scanner = new Scanner(  
    System.in);  
String s = scanner.next();  
int i = scanner.nextInt();
```



# Alternatives

**ConsoleProgram**

**DialogProgram**

**GraphicsProgram**

# Java AWT

This is the original Java GUI library. Very outdated. You shouldn't use this one.

# Java Swing

The second Java GUI library. Quite flexible and different visual styles available. Can still be used, but has been superseded.

Web resource:

[Creating a GUI With JFC/Swing](#)

# JavaFX

The current Java GUI library. Themeable, support for animations. This is a good choice for your applications.

Web resource:

[Java Client Technologies](#)