# CS18000: Problem Solving and Object-Oriented Programming

## Simple GUIs

# Video 1
# JOptionPane Class

# Simple Graphical User Interfaces

Dialogs

# Text-Based Interface vs. GUI

- Text-Based Interface
  - Program prompts user for data
  - User enters data via keyboard
  - Output in a “terminal window”
  - Old days: the terminal was the window
- GUI (Graphical User Interface)
  - Window displays set of “controls” or “widgets”
  - User interacts with controls
  - Program responds to user “events” with actions

# Dialog Concepts

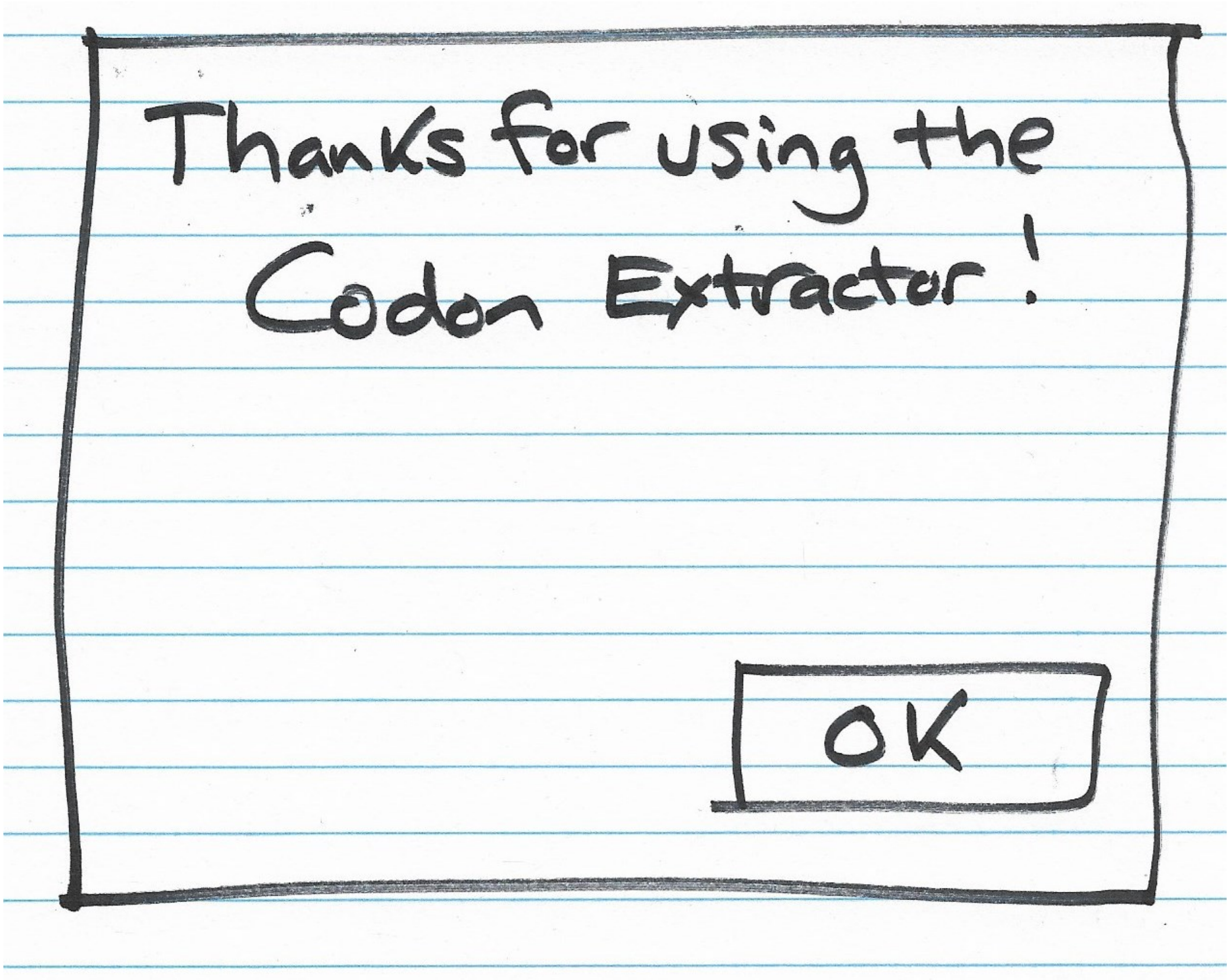
- Prerequisite: application must be running on JVM with window system configured
- Java dialogs are "modal"
  - Application code (your program) "blocks" waiting for user response
  - Similar to using Scanner to read from keyboard
- Java GUI components adapt to "Look and Feel" of local system

# JOptionPane Class

- Java workhorse for modal dialogs
- Part of Java GUI package: “Swing”

```
import javax.swing.JOptionPane;
```

- Several static methods for typical use cases…
  - showMessageDialog
  - showInputDialog
  - showConfirmDialog
  - showOptionDialog



Thanks for using the Codon Extractor!

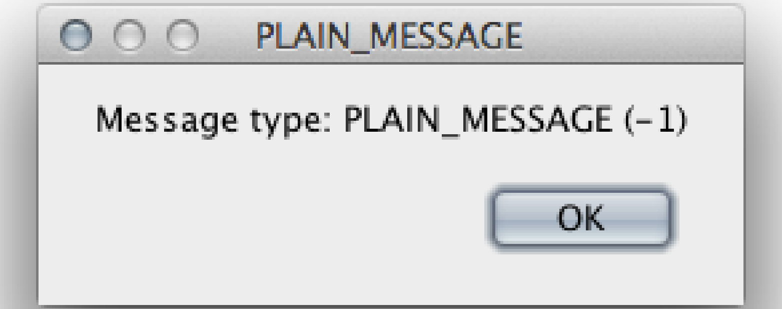
# JOptionPane Class

- Common arguments…
  - Location where dialog pops up (null is center screen)
  - Message to be included in dialog box (may be string or icon or html)
  - Message type (used for “look and feel” and helpful icon)
  - Option type (what buttons should be included by default)
  - Options (e.g., array of Strings for button names)
  - Icon to replace default icon of message type
  - Title string to be used in window heading
  - Initial value (a default value for certain option types)
- Many arguments can be omitted for default values

# Message Type Parameter

- Message Type selects icon to display
- Look and Feel dependent
- Possible values
  - JOptionPane.PLAIN_MESSAGE (-1)
  - JOptionPane.ERROR_MESSAGE (0)
  - JOptionPane.INFORMATION_MESSAGE (1)
  - JOptionPane.WARNING_MESSAGE (2)
  - JOptionPane.QUESTION_MESSAGE (3)

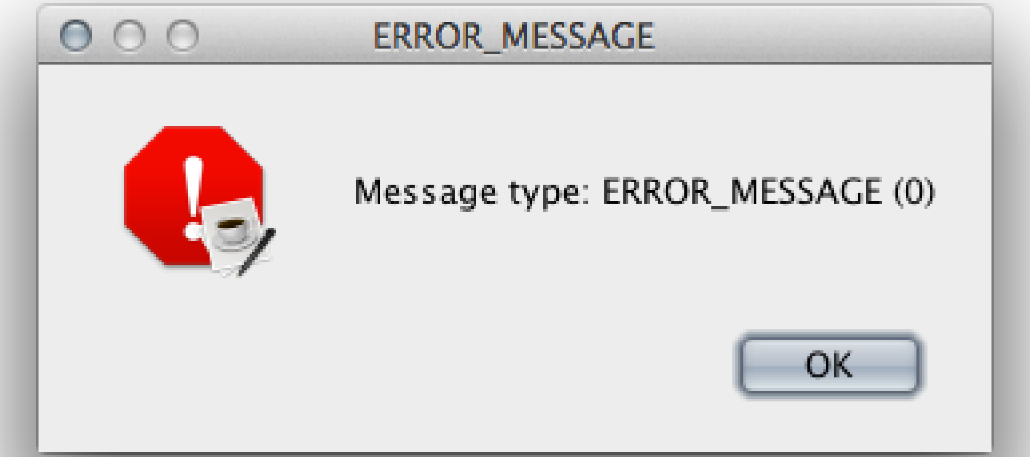
# JOptionPane.PLAIN_MESSAGE



PLAIN_MESSAGE

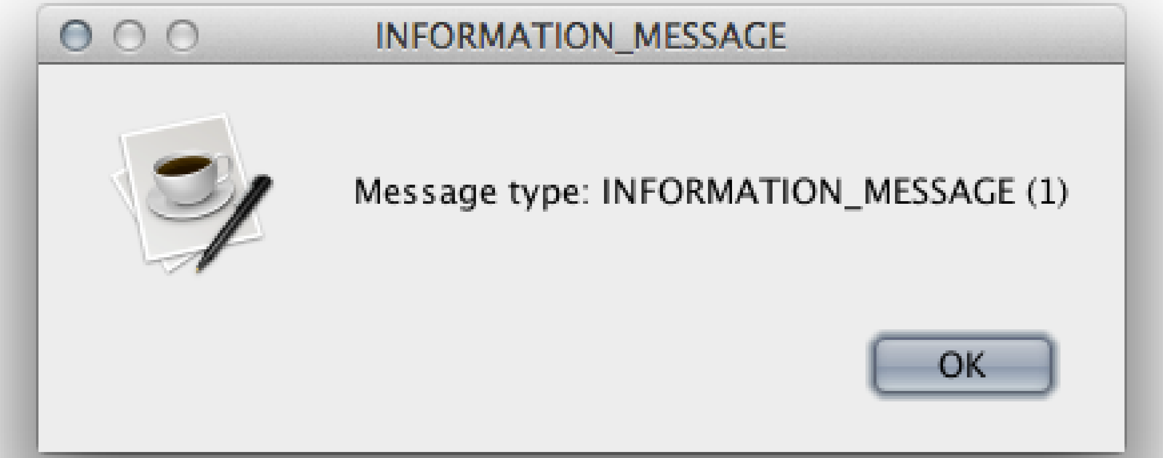
Message type: PLAIN_MESSAGE (-1)

# JOptionPane.ERROR_MESSAGE



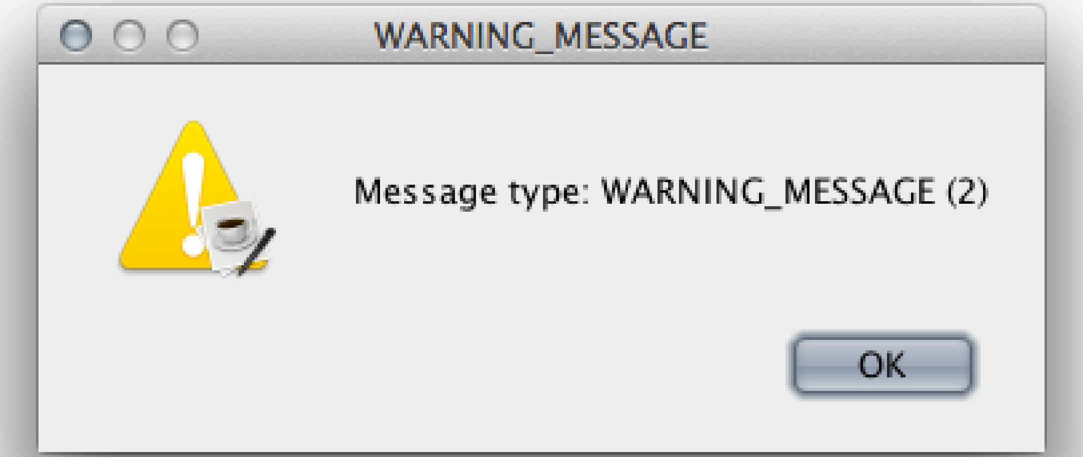
Message type: ERROR_MESSAGE (0)

# JOptionPane.INFORMATION_MESSAGE



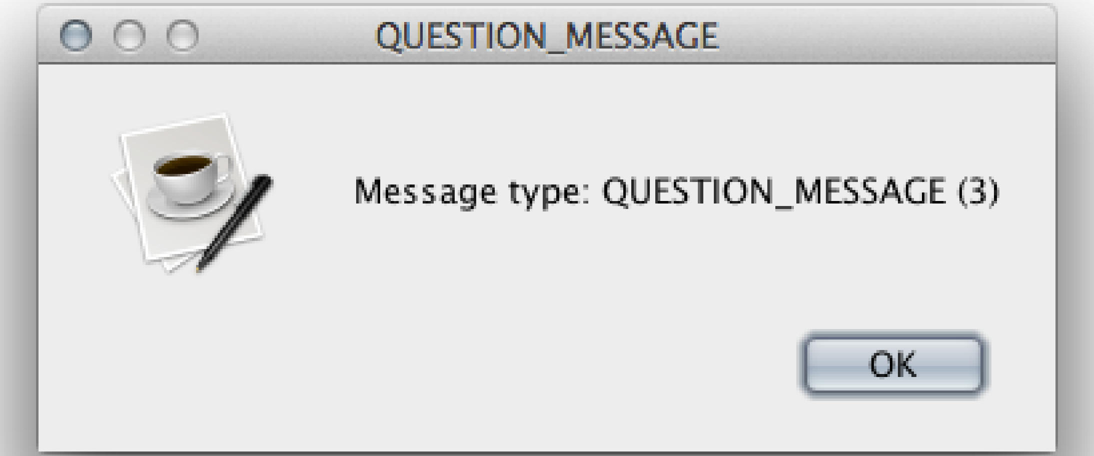
Message type: INFORMATION_MESSAGE (1)

# JOptionPane.WARNING_MESSAGE



Message type: WARNING_MESSAGE (2)

# JOptionPane.QUESTION_MESSAGE



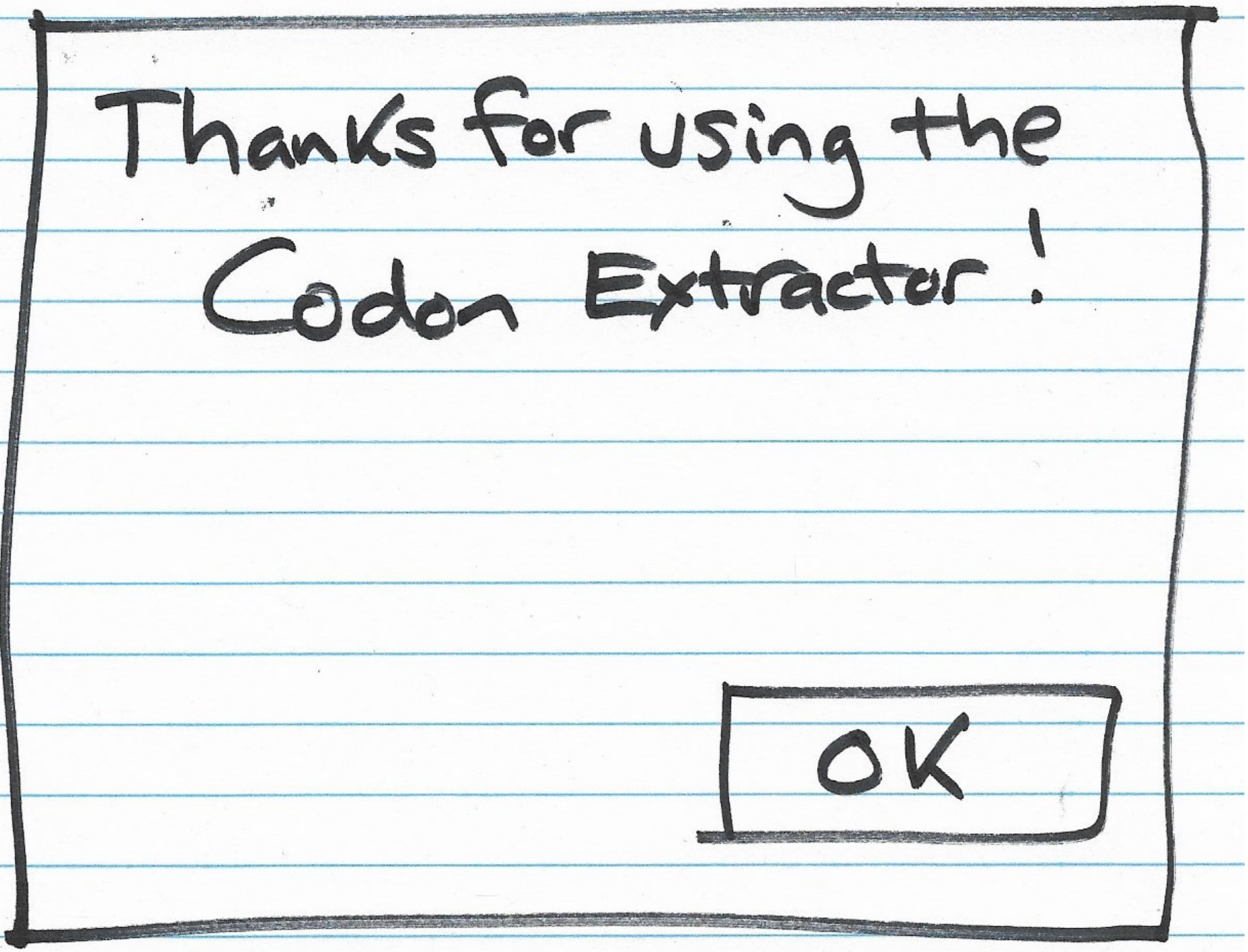
QUESTION_MESSAGE

Message type: QUESTION_MESSAGE (3)

# Video 2
# JOptionPane Methods

# showMessageDialog

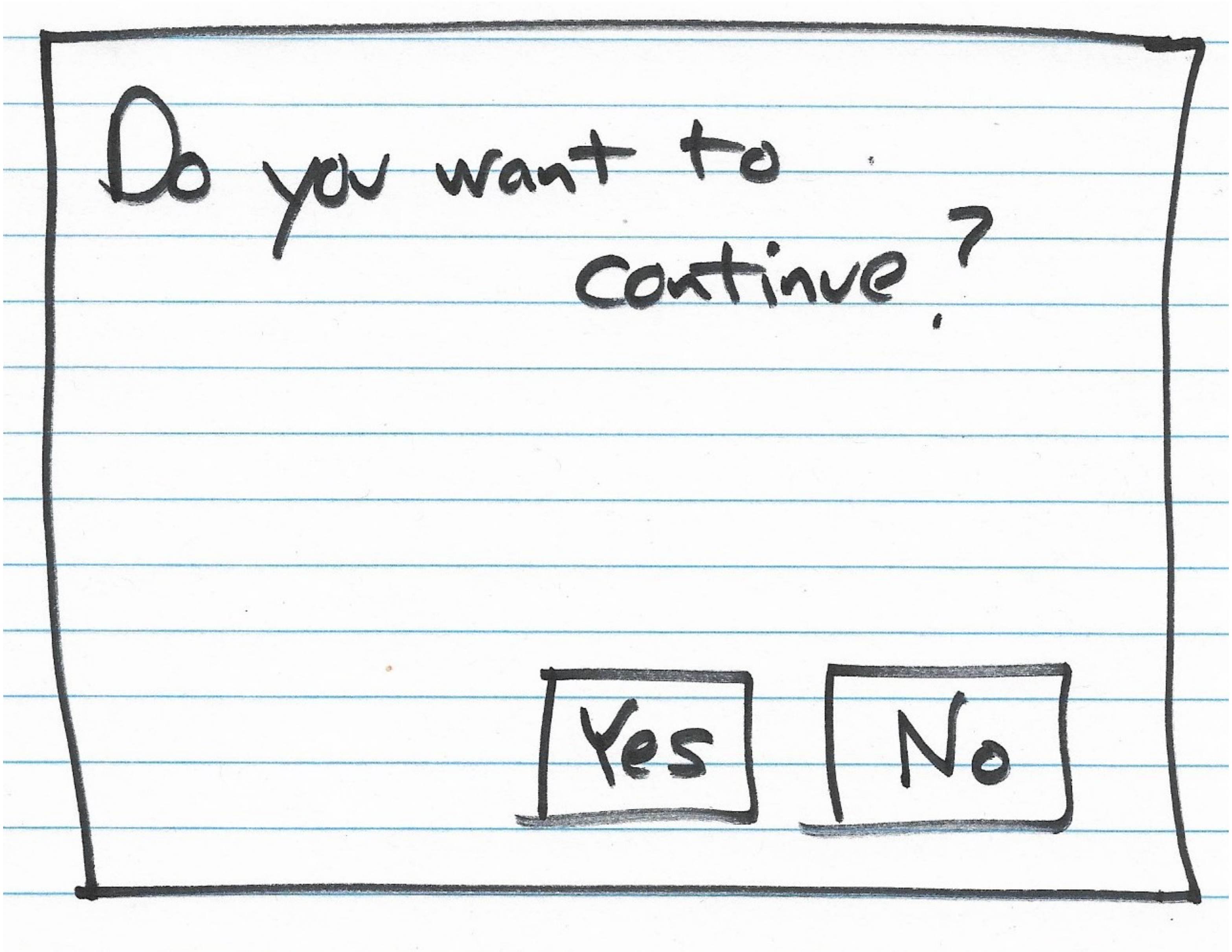
- Simplest dialog
- At minimum, displays message to user
- Can include other parameters to affect appearance
- Only one of these methods with a void return value—it is a do-only method



Thanks for using the Codon Extractor!

# showConfirmDialog

- Asks the user to confirm an action
- Default options: “Yes”, “No”, “Cancel”
- Returns int value indicating which button user selected
- Various button combinations available…
  - “Yes” or “No”
  - “OK” or “Cancel”
  - Or user configurable with list of Strings



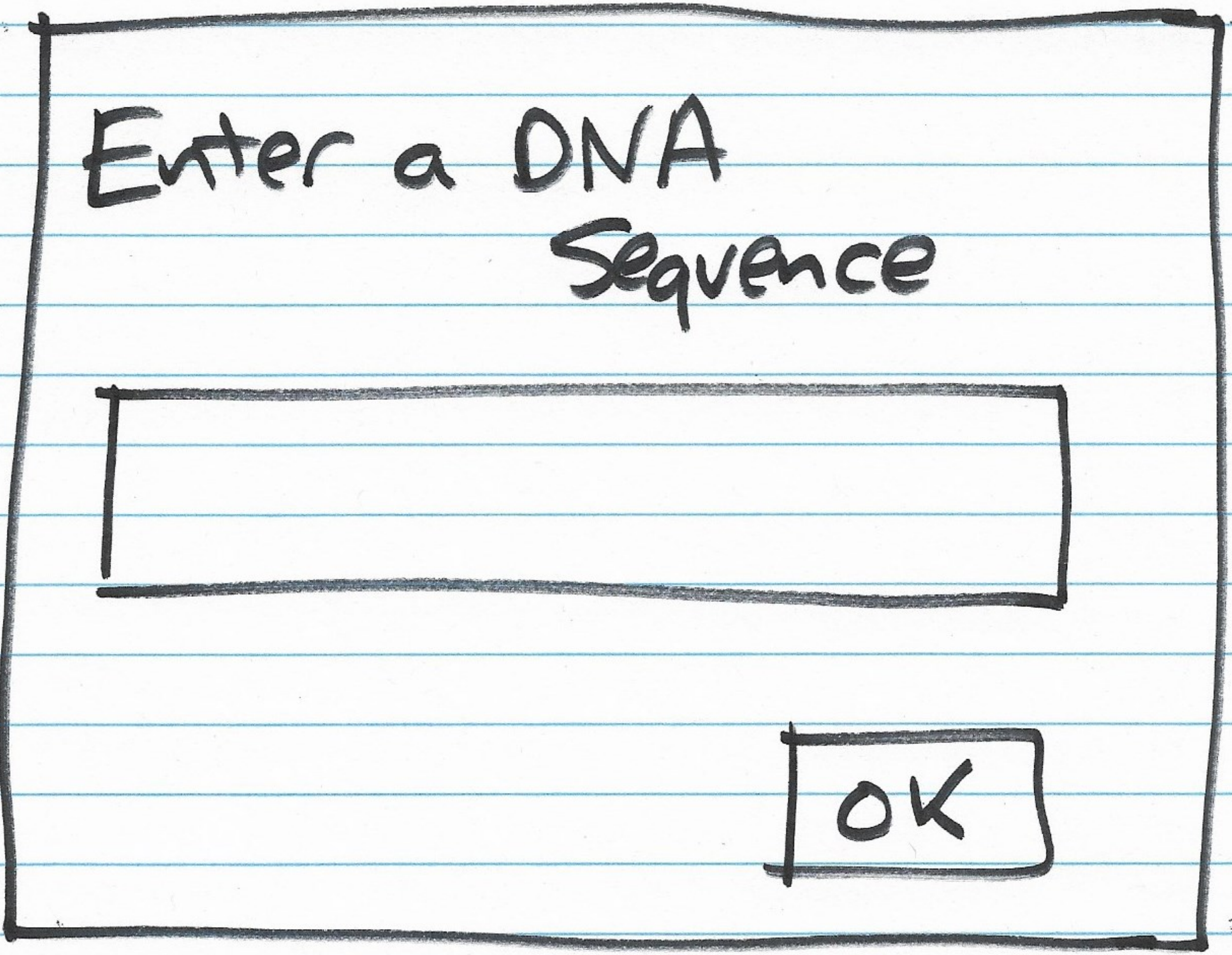
Do you want to continue?

# Values with showConfirmDialog

- Parameter option types...
  - – JOptionPane.YES_NO_OPTION
  - – JOptionPane.YES_NO_CANCEL_OPTION
  - – JOptionPane.OK_CANCEL_OPTION
- Returns one of…
  - – JOptionPane.YES_OPTION (0) (same as OK_OPTION)
  - – JOptionPane.NO_OPTION (1)
  - – JOptionPane.CANCEL_OPTION (2)
  - – JOptionPane.CLOSED_OPTION (-1)

# showInputDialog

- Asks the user for some input
- Returns String value
- Input may be…
  - Freely typed text
  - Selected from drop-down box or list
- Allows simplified arguments
- To create a drop-down box or list…
  - Provide array of Strings and default value
  - Must cast return value to String



Enter a DNA Sequence

# showOptionDialog

- Generalized version: configurable buttons
- Returns index of button selected
- Way too many parameters…
  - Component parentComponent
  - Object message
  - String title
  - int optionType
  - int messageType
  - Icon icon
  - Object[] options
  - Object initialValue

# Dialog Demo Code

- Available here: http://bit.ly/WeYeZC

# Video 3
# CodonExtractor Example

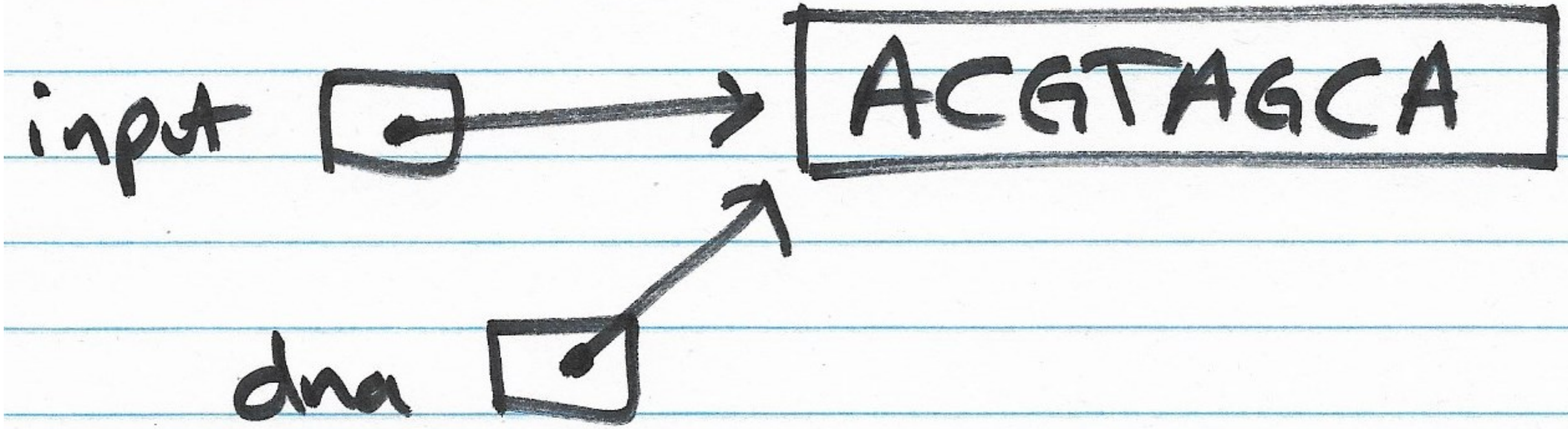
# Simple Graphical User Interfaces

GUI Examples

# Problem: CodonExtractor

- Write a program that reads a DNA sequence from the user and displays the codons in it
- Definitions:
  - DNA sequence: sequence of chars in ACGT
  - Codon: sequence of three chars in DNA sequence
- Algorithm:
  - Prompt user for DNA, check for valid input
  - Break DNA into 3-character chunks, display
  - Repeat until user indicates done

ACGTAGCA



# CodonExtractor: main Method

```
int continueProgram;

do {
    // Read DNAsequence
    String input = JOptionPane.showInputDialog("Enter a DNA sequence");
    input = input.toUpperCase(); // Make upper case
    String message = "Do you want to continue?";

    if (isValid(input)) // Check for validity
        displayCodons(input); // Find and display codons
    else
        message = "Invalid DNA Sequence.\n" + message;

    continueProgram = JOptionPane.showConfirmDialog(null, message,
                                                    "Alert",
                                                    JOptionPane.YES_NO_OPTION);
    } while (continueProgram == JOptionPane.YES_OPTION);

JOptionPane.showMessageDialog(null, "Thanks for using the Codon Extractor!");
```

# CodonExtractor: isValid

```
public static boolean isValid(String dna) {
    String validBases = "ACGT";

    for (int i = 0; i < dna.length(); i++) {
        char base = dna.charAt(i);
        if (validBases.indexOf(base) == -1)
            return false; //base not in "ACGT"
    }
    return true;
}
```

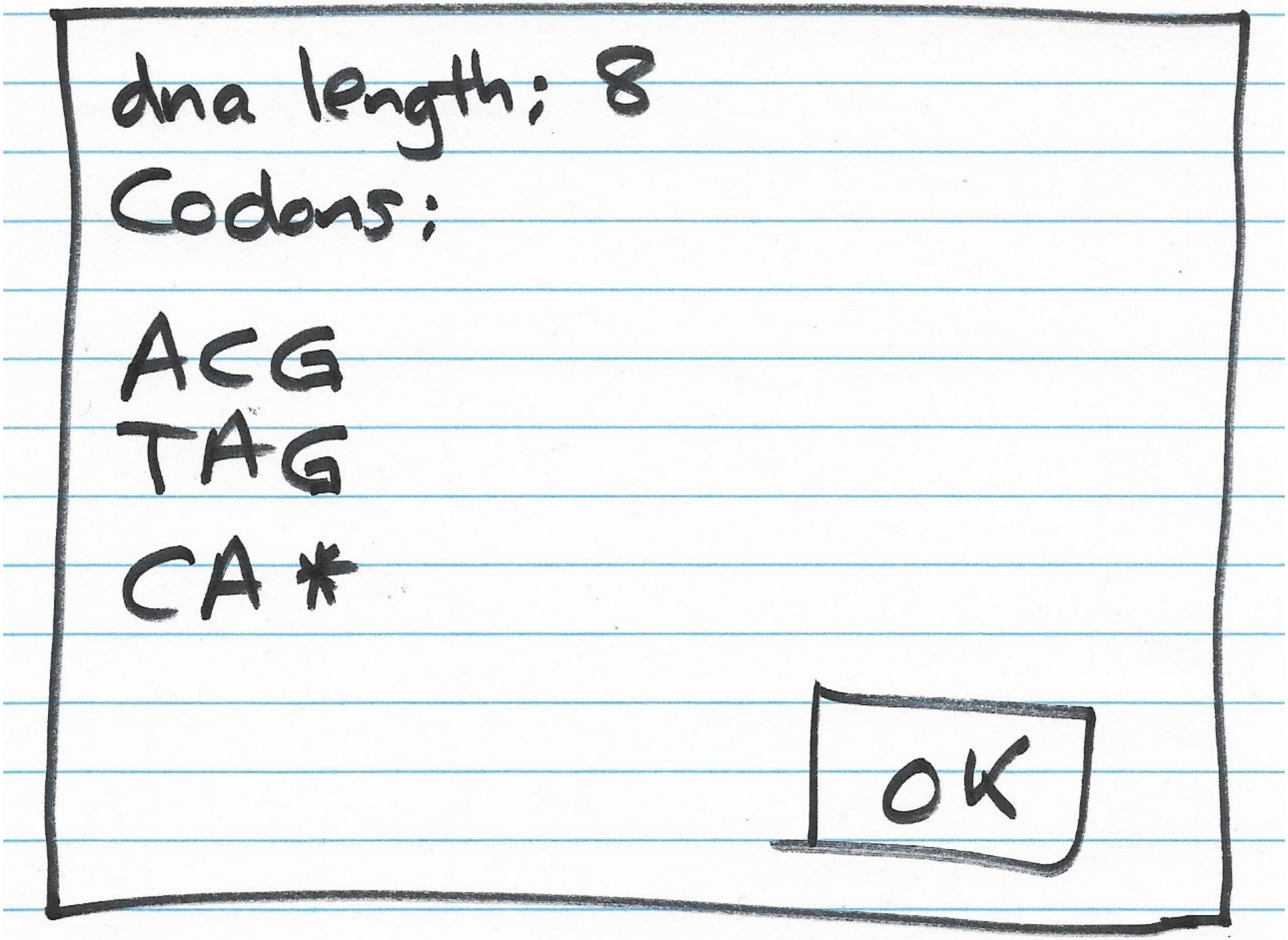
# CodonExtractor: displayCodons

```
public static void displayCodons(String dna) {
    String message = "";

    // Get as many complete codons as possible
    for (int i = 0; i < dna.length() - 2; i += 3)
        message += "\n" + dna.substring(i, i + 3);

    // 1-2 bases might be left over
    int remaining = dna.length() % 3;
    if (remaining == 1)
        message += "\n"+ dna.substring(dna.length() - 1,
                                       dna.length()) + "**";
    else if (remaining == 2)
        message += "\n"+ dna.substring(dna.length() - 2,
                                       dna.length()) + "*";

    message = "dna length: " + dna.length() + "\n\nCodons: " + message;
    JOptionPane.showMessageDialog(null, message,
                                  "Codons in DNA",
                                  JOptionPane.INFORMATION_MESSAGE);
}
```

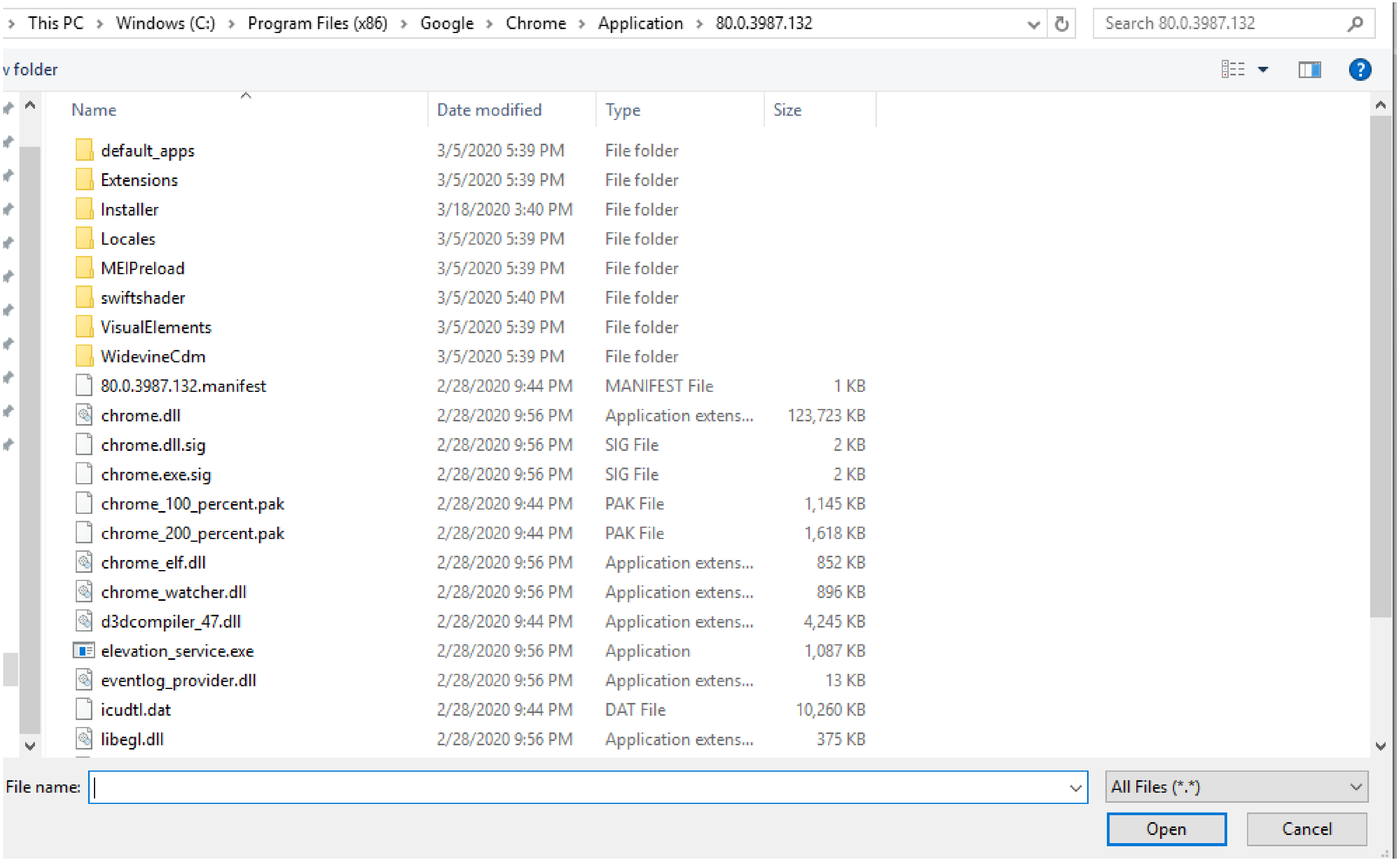


dna length: 8

Codons:

ACG
TAG
CA*

# Video 4
# JFileChooser



This PC › Windows (C:) › Program Files (x86) › Google › Chrome › Application › 80.0.3987.132

| Name | Date modified | Type | Size |
|---|---|---|---|
| default_apps | 3/5/2020 5:39 PM | File folder | |
| Extensions | 3/5/2020 5:39 PM | File folder | |
| Installer | 3/18/2020 3:40 PM | File folder | |
| Locales | 3/5/2020 5:39 PM | File folder | |
| MEIPreload | 3/5/2020 5:39 PM | File folder | |
| swiftshader | 3/5/2020 5:40 PM | File folder | |
| VisualElements | 3/5/2020 5:39 PM | File folder | |
| WidevineCdm | 3/5/2020 5:39 PM | File folder | |
| 80.0.3987.132.manifest | 2/28/2020 9:44 PM | MANIFEST File | 1 KB |
| chrome.dll | 2/28/2020 9:56 PM | Application extens... | 123,723 KB |
| chrome.dll.sig | 2/28/2020 9:56 PM | SIG File | 2 KB |
| chrome.exe.sig | 2/28/2020 9:56 PM | SIG File | 2 KB |
| chrome_100_percent.pak | 2/28/2020 9:44 PM | PAK File | 1,145 KB |
| chrome_200_percent.pak | 2/28/2020 9:44 PM | PAK File | 1,618 KB |
| chrome_elf.dll | 2/28/2020 9:56 PM | Application extens... | 852 KB |
| chrome_watcher.dll | 2/28/2020 9:56 PM | Application extens... | 896 KB |
| d3dcompiler_47.dll | 2/28/2020 9:44 PM | Application extens... | 4,245 KB |
| elevation_service.exe | 2/28/2020 9:56 PM | Application | 1,087 KB |
| eventlog_provider.dll | 2/28/2020 9:56 PM | Application extens... | 13 KB |
| icudtl.dat | 2/28/2020 9:44 PM | DAT File | 10,260 KB |
| libegl.dll | 2/28/2020 9:56 PM | Application extens... | 375 KB |

File name:

All Files (*.*)

# Problem: Prompting for a File Name

- JFileChooser (javax.swing.JFileChooser)
- Use new to create an object
- Set title bar with setDialogTitle(title)
- Show with showOpenDialog(null)
- Return value is an int: 0 open, 1 cancel
- Get the File selected with getSelectedFile()
- File object describes the name and location of (the path to) the file

# Solution: Prompting for a File Name

```
import java.io.File;
import javax.swing.JFileChooser;

public class FileChooser {
    public static void main(String[] args) {
        JFileChooser fc = new JFileChooser();

        fc.setDialogTitle("Choose Important File");
        int val = fc.showOpenDialog(null);
        System.out.println(val);

        File f = fc.getSelectedFile();
        System.out.println(f);
    }
}
```