13: Secret Decoders

Encryption

Write the Secret Messenger Code

Start the GUI

Review

Review

Encryption

What is Encryption?

- Encryption involves taking plain text and translating it into a form that people cannot easily read.
- It also involves taking the text that cannot be easily read and translating it back into plain text.
- Encryption is used in banking, government, covert agencies, and military.

Rot 13

This is a simple encryption standard that uses a little bit of math.

- Take each letter of the alphabet and assign it a number.
  - A is 0. (We always begin counting with zero in computer science.)
  - Z is 25.
- There are 26 letters in the alphabet and half of that is 13. This is where the name "Rot 13" comes from.

How to encrypt.

- Convert all letters to upper case.
- If a character isn't a letter (such as a number or a period), we ignore it.
- Convert each letter to a number. (A=0, B=1, C=2, . . . , Z=25).
- If a number is greater than or equal to 13, subtract 13.
- If a number is less than 13, add 13.
- Convert the numbers back to letters.

Example: TABLE

Let's take the word "TABLE".

- This converted into numbers is [19, 0, 1, 11, 4].
- 19 is greater than or equal to 13, so we find "19 - 13", which is 6.
- We add 13 to the other numbers, which will be 13, 14, 24, and 17.
- We convert those numbers back into letters.
- Our encrypted word is "GNOYR".

Example 2: GNOYR.

In order to convert our encrypted word back into the original, we do the same steps.

- We already know that "GNOYR" becomes [6, 13, 14, 24, 17].
- If a number is less than 13, add 13.
  - 6 becomes 19.
- If a number is greater than or equal to 13, subtract 13.
  - [13, 14, 24, 17] is now [0, 1, 11, 4]
- Convert those numbers back to letters.
  - [19, 0, 1, 11, 4] is "TABLE".

Write the Secret Messenger Code

Let's get started.

- Make a new project called "SecretMessages".

Extend as an Application.

We've got to do some setup first. Find the line that looks like this:

```
public class SecretMessages {
```

Change it to this:

```
public class SecretMessages extends Application {
```

Modify Main

Add this line to `main`. We had to do this for our Turtles.

```
public static void main(String[] args) {
    launch(args);
}
```

Errors.

- There is still an error in the code. JavaFX requires a method to work, but NetBeans will introduce it for us.
- The word "SecretMessages" will have a red squiggly line under it.
- Click the word with your mouse once so that the cursor is in the word.
- Hit "ALT+Enter" on the keyboard.
- Select the first option "Implement all abstract methods."

Modify the method.

- You should find a new method in your code. NetBeans added this for us.
- In the method, you'll find this line.

Code.

```
throw new UnsupportedOperationException("...");
```

Remove this line. NetBeans wants you to remove it. Your program will crash if you keep it. Leave `start` for now.

Part 1: Write a new method `encrypt`.

This method will take a single string an build an encrypted message.

```
public String encrypt(String original) {
    String encrypted = "";

    for (int i = 0; i < original.length(); i++) {
        char ch = original.charAt(i);
```

Part 2: Encrypt the message.

This if statement does the job of converting the characters to numbers, performing the math, and building the encrypted string.

```
if (Character.isUpperCase(ch)) {
    int x = (int)(ch - 'A');

    if (x >= 13)
        x -= 13;
    else
        x += 13;

    encrypted += (char) (x + 'A');
}
```

Part 3: Encrypt the message.

This else branch is for the if statement on the previous slide.

```
        else
            encrypted += ch;
    } // End For

    return encrypted;
} // End encrypt
```

Start the GUI

Time to start the GUI

- Unlike the previous game, there's only three GUI elements that we need:
  - A TextArea for input.
  - A Button to perform an action.
  - A TextArea for output.

Part 1: Add some code to `start`.

These first few elements build the scene.

```
TextArea input = new TextArea();
TextArea output = new TextArea();
Button button = new Button("Encrypt!");

VBox vPane = new VBox();
vPane.getChildren().addAll(input, button, output);
```

Part 2: Establish the scene and show.

```
Scene scene = new Scene(vPane);
primaryStage.setScene(scene);
primaryStage.show();
```

Part 3: Give the Button an action and go!

```
        button.setOnAction(event -> {
                String original = input.getText().toUpperCase();
                output.setText(encrypt(original));
            });
    } // End start
```

Review

Review