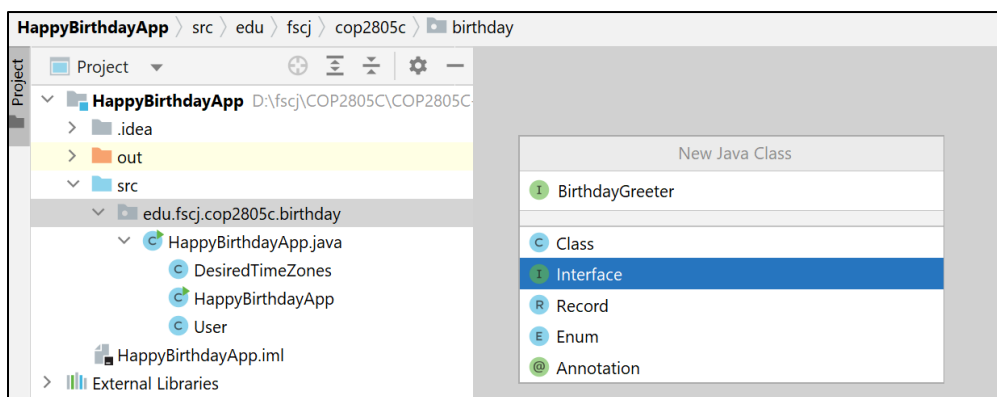
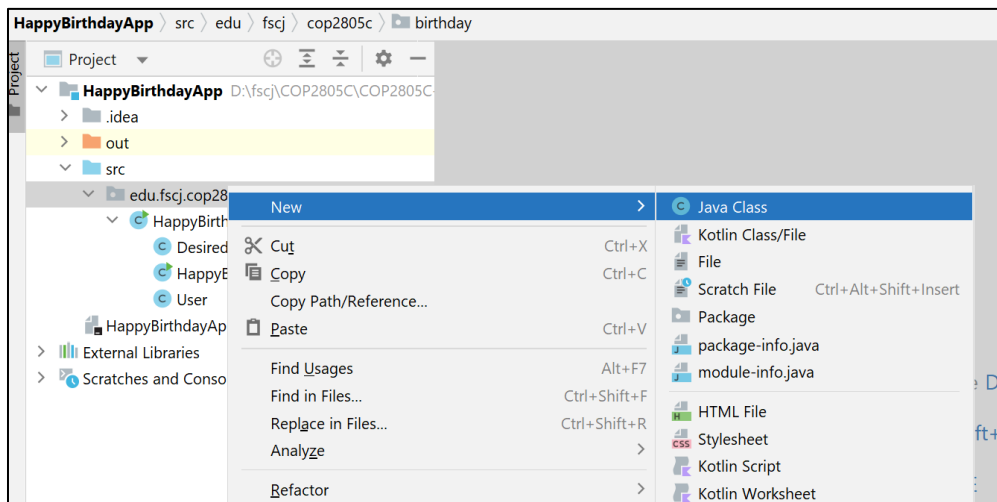


COP3330C Module 3 Practice Exercise

For this practice exercise we will do the following:

- Implement a varargs method (also known as "variable arity") as discussed in Chapter 5 of our textbook (Methods). Note that variable arity is used in some Java APIs that we are already familiar with (e.g. the `System.out.printf` method).
 - Using the varargs method, add a new feature to the Happy Birthday App from the Module 2 practice exercise which uses a loop to allow the user to enter multiple birthdays. The `HappyBirthdayApp` class should store these in an `ArrayList` (the "birthday list").
 - Add a method with variable arity which accepts a variable number of `User` objects and adds them to the birthday list. This variable arity method "addBirthdays" can be found in the `HappyBirthdayApp` class in the solution for this exercise.
- Create an interface and implement the abstract methods in an application. We will also use the `StringBuilder` to help us craft formatted text.
 - Add a "BirthdayGreeter" interface to your project. The following images demonstrate how to do this in IntelliJ:



- Add two abstract methods to the interface:

```

1 package edu.fscj.cop2805c.birthday;
2
3 public interface BirthdayGreeter {
4     // build a birthday card
5     // 1 implementation
6     public String buildCard(String msg);
7     // send a birthday card
8     // 1 implementation
9     public void sendCard(User u);
10

```

- Modify the HappyBirthdayApp class to implement the interface:

```

// main mpplication class
public class HappyBirthdayApp implements BirthdayGreeter {

```

- Implement the abstract methods in the code to produce the output shown below (names and greeting messages can vary based on your preferences).
- The buildCard method builds a birthday card formatted as a String with the outline characters shown in the output, given a (possibly multi-line) message provided by the program.
- The sendCard method calls the buildCard method and then "sends" the card via email to the user (it doesn't actually send it, but there are APIs we could use to do this!)

In my solution you will see that I removed the desired time zone ArrayList from the Happy Birthday App and encapsulated it in its own class (DesiredTimeZones). The showTZMenu method was moved from the HappyBirthdayApp class into this new class.

I have also removed the interactive portion of the application and restricted it to use only program-generated test data.

Sample output (3 test users - there is no user interaction in this version of the application):

Here are the birthdays:
Dianne Romero:
Sorry, today is not their birthday.

Sally Roberts:

```

|-----|
|      Happy Birthay, Sally Roberts      |
| Hope all of your birthday wishes come true! |
|-----|

```

sending email to Sally.Roberts@email.test

Edwin Peterson:

```
|-----|  
|      Happy Birthay, Edwin Peterson      |  
| Hope all of your birthday wishes come true! |  
|-----|
```

sending email to Edwin.Peterson@email.test