

COP3330C Module 6 Graded Assignment

In this graded assignment we will complete an implementation of the Birthday Card application using starter files provided in the GitHub repository. This new design applies the **Abstract Factory** pattern to the creation of two different presentation types of birthday card greetings: standard text-based greetings and graphical (image-based) greetings.

Look for the comments containing numbered “FIB” (fill-in-the-blank) headings in the following source code files:

```
BirthdayCard.java  
BirthdayCardFactory.java  
HappyBirthdayApp.java
```

There are 6 sections that must be completed in order for the application to execute properly.

After adding the missing code, execute the program and copy your output into a separate Word document.

Submit your modified source code and Word output to the GitHub repository.

The following notes describe the application design and changes made to the Module 5 Practice Exercise.

For our design we will use the BirthdayCardFactory as our top-level abstract factory. There will be two concrete factories that extend the abstract factory, one factory to produce textual greetings and one for the graphical versions. Each factory will produce age-appropriate versions of the factory-specific type of birthday card (text or image).

For the graphical greetings we will just “pretend” to send these by showing the file name of the image. The age-specific image files are stored in the “resources” folder, parallel to the src folder; this is a standard way to structure non-code resources for an application. We could use a graphics framework (e.g. JavaFX) to actually show these images.

We will make our BirthdayCard class abstract, but the age category subclasses will still continue to inherit from this class. The two factories will generate appropriate greetings based on the age category, but each factory will produce a different presentation type.

The following changes have been made:

- Add two concrete factory classes that extend the abstract BirthdayCardFactory class which will create age-appropriate text-based or image-based cards. In each of these classes add four methods to create the age-appropriate card. The BirthdayCardSender parameter has been removed from these, we have moved the responsibility for sending the card back to the client (HappyBirthdayApp) class. Each of these methods, when implemented in the factory subclasses, will return a Birthday card of the appropriate type (text or image) for the appropriate age class.

```
public void createCardChild(User u)  
public void createCardAdolescent(User u)
```

```

    public void createCardAdult(User u)
    public void createCardSenior(User u)

```

- Split the BirthdayCard class and its subclasses into a separate file: "BirthdayCard.java"
- Declare the BirthdayCard class as abstract
- Change the existing four BirthdayCard subclasses so they generate text-based cards (e.g. add Text to the end of the class name). Move the card greeting message from the Module 5 factory subclasses to these subclasses. Here is the subclass representing a child:

```

class BirthdayCard_ChildText extends BirthdayCard {
    private static final String MSG =
        """
        Happy Birthday! Hope your day is filled with magic, fun,
        and all your birthday wishes coming true! 🎂
        """;
    public BirthdayCard_ChildText(User u) {
        super(u, MSG);
    }
}

```

- Add an Image class and ImageType class to the project. The image class contains the image file name and image type (PNG or JPG) for each card
- Add the image files to a new resources folder
- Add an overloaded constructor to the abstract BirthdayCard class which includes an Image:

```

    public BirthdayCard(User user, Image cardImage)

```

- Add an image member to the abstract BirthdayCard class. The class will use this member for image-based cards or the msg member for text-based cards, but not both.
- Create four new BirthdayCard subclasses that generate image -based cards (copy/paste the Text subclasses and change "Text" to "Image"). Here is the card subclass for a child:

```

class BirthdayCard_ChildImage extends BirthdayCard {
    private static final Image IMAGE = new Image(
        BirthdayCardFactoryImage.getImageFile(
            ImageType.JPG, "child"),
        ImageType.JPG);
    public BirthdayCard_ChildImage(User u) {
        super(u, IMAGE);
    }
}

```

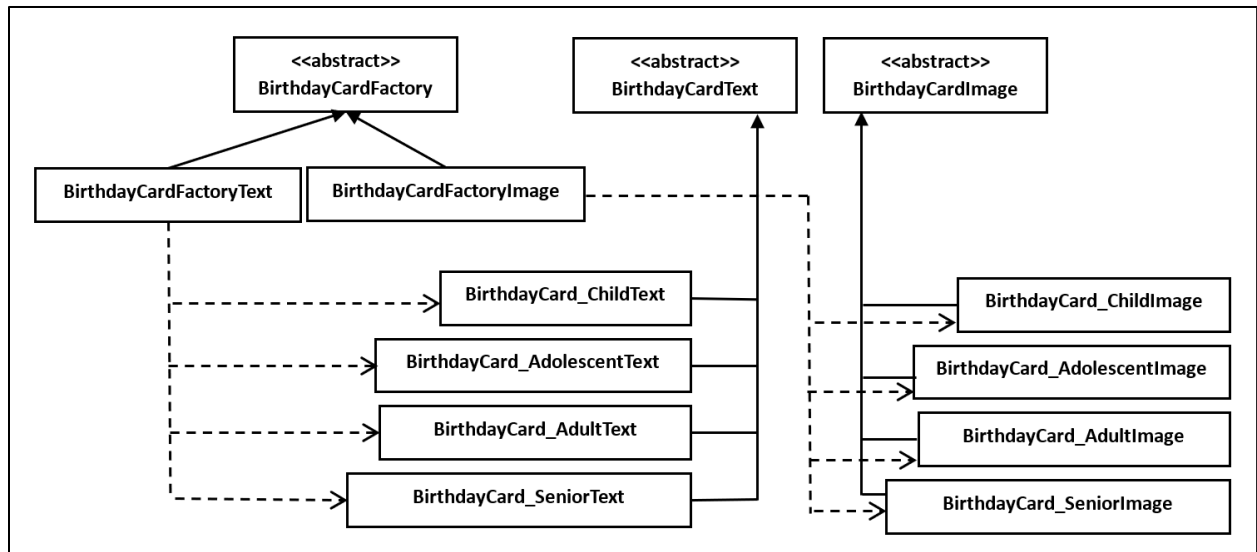
The unit test in the application class now loops through the users and alternates between text- and image-based cards. For text-based cards, the text-based factory subclass is instantiated, the image-based factory is used for image-based cards.

```
    BirthdayCard card = null;
    count++;

    // alternate between text and graphics card factories
    cardFactory = (count % 2 == 0) ?
        new BirthdayCardFactoryText() :
        new BirthdayCardFactoryImage();

    switch (count) {
        case 1:
            // create a card for child
            card = cardFactory.createCardChild(u);
            break;
        case 2:
            // create a card for an adolescent
            card = cardFactory.createCardAdolescent(u);
            break;
        case 3:
            // create a card for an adult
            card = cardFactory.createCardAdult(u);
            break;
        case 4:
            // create a card for a senior
            card = cardFactory.createCardSenior(u);
            break;
    }
```

The following image represents the UML class diagram for the application (client is not shown)



Sample Output (I have added a debugging println in the BirthdayCard constructor to indicate which card subclass being generated).

```
Here are the birthdays:
Miles Bennell:
Sorry, today is not their birthday.

Becky Driscoll:
Birthday card for Becky Driscoll
bdc-child.jpg (JPG)

sending email to Becky.Driscoll@email.test

Jack Belicec:
Birthday card for Jack Belicec
|-----|
| Happy Birthday! Hope your day is as awesome as you are |
|           and all your birthday wishes come true! 🎉🎊           |
|-----|

sending email to Jack.Belicec@email.test

Theodora Belicec:
Birthday card for Theodora Belicec
bdc-adult.jpg (JPG)

sending email to Theodora.Belicec.@email.test

Sally Withers:
Birthday card for Sally Withers
|-----|
|           Happy Birthday!           |
| May this special day bring you cherished memories, joy |
|           in the present, and hopes for the future. |
| Wishing all of your birthday dreams and wishes come true! |
|-----|

sending email to Sally.Withers@email.test
```