Homework 07: Interfaces

Release date: 2/25

Due date: 11:59PM, 3/4

Goals:

- Become familiar with declaring interfaces
- Practice implementing an interface with a class
- Practice working with enum declarations and values

Introduction:

Interfaces are an important part of many APIs (Application Programming Interfaces) in Java. They provide a way to make a group of classes have a common supertype. An interface can really be thought of as a contract — any class that implements an interface makes a promise that it will provide implementations for the methods declared in it.

Description

For this homework, your task will be to create **`PresidentialCandidate`** and **`SenatorialCandidate`** classes in order to store information about political candidates and, most importantly, check if they can run for President and Senator, respectively, of the United States. To do this, our classes will implement an **`Electable`** interface which lays out some requirements for being elected.

`Electable interface`	
`String getName()`	Returns the name of the candidate
`boolean isCitizen()`	Returns the citizenship status*
`int getAge()`	Returns the age of the candidate
`String getPartyMembership()`	Returns the candidate's claimed party
`CampaignIssue getCampaignFocus()`	Returns the `CampaignIssue` the candidate's campaign focuses on
`boolean canBeElected()`	Determines if a candidate can be elected and returns the result. To be elected, one must be a named citizen of **at least the minimum age** and **have a non-NULL claimed party**.

** citizenship status includes requirements such as being a natural-born citizen of residence of 14+ years if running for President, unrelated to your implementation*

What is the minimum age to be elected? Well, it depends on the position you're running for!

Minimum Age to Run for US Offices	
Position	**Minimum Age**
President	35
Senator	30

You'll need this later when we begin to code the methods in PresidentialCandidate.java and SenatorialCandidate.java!

In order to represent the possible campaign issues the candidates might focus on, we will be using an **`enum`** titled **`CampaignIssue`** (as used by **`Electable`**). Create a new file titled ***CampaignIssue.java*** and insert an **`enum`** declaration like so:

```
public enum EnumName {
	FIRST_VALUE,
	SECOND_VALUE,
	.
	.
	.
	LAST_VALUE
}
```

In our case, **`EnumName`** will be replaced with **`CampaignIssue`**, and the values inside the braces can be found in the next table.

Enums are handy because they allow us to use more informative names for values than just numbers! For example, instead of saying that our candidate's campaign focuses on issue 1, we can say that the candidate's campaign focuses on **`HEALTHCARE`**.

An enum allows us to define multiple constants at once which are internally assigned values by java. We can then compare those values without worrying about the numbers at all! Therefore, we can use the enum constants in if statements and comparisons. We can check if a candidate has a specific campaign focus like so:

```
	if (washington.campaignFocus == RENEWABLE_ENERGY) {
		System.out.println("Washington's campaign focuses on
			Renewable Energy!");
	}
```

For more examples, https://docs.oracle.com/javase/tutorial/java/javaOO/enum.html has a few nice examples with enums. You might find them when working with days of the week or months of the calendar year. Days like **`MONDAY`**, **`TUESDAY`**, and **`SUNDAY`** are easier than days 0, 1, 2, etc. We don't need to worry if the week starts with Monday or Sunday when using enums!

Below are the campaign issues we will be including. The numbered order should correspond to the order they are created in ***CampaignIssue.java***.

CampaignIssue	
0. NO_FOCUS	5. EDUCATION
1. HEALTHCARE	6. RENEWABLE_ENERGY
2. GUN_CONTROL	7. TAX_REFORM
3. CIVIL_RIGHTS	8. IMMIGRATION
4. CLIMATE_CHANGE	9. THE_ECONOMY

Now, let's shift our attention to the **`PresidentialCandidate`** class. Recall that **`PresidentialCandidate`** will implement our **`Electable`** interface. This means that **`PresidentialCandidate`** will need to implement each method in **`Electable.java`**. In addition, we will add a couple constructors and a method to update the candidate's campaign focus.

PresidentialCandidate class	
`PresidentialCandidate(String name, boolean citizenship, int age, String party)`	Typical constructor for a `PresidentialCandidate` object; `campaignFocus` will default to `NO_FOCUS`
`PresidentialCandidate(String name, boolean citizenship, int age, String party, CampaignIssue campaignFocus)`	Alternate constructor for a `PresidentialCandidate` that sets the `campaignFocus` attribute
`void updateCampaignFocus (CampaignIssue issue)`	Update's the candidate's `campaignFocus` to `issue`

`SenatorialCandidate` is setup very similarly to **`PresidentialCandidate`**. Begin by implementing **`Electable`**. Then, add the same 3 additional methods we added to **`PresidentialCandidate`** - (two constructors and an **`updateCampaignFocus()`** method).

Sample output

As always, we recommend testing your code (in IntelliJ first, not in Vocareum)! A sample `main ()` method may look like the following:

```
public static void main(String[] args) {
      PresidentialCandidate trump = new PresidentialCandidate("Donald Trump",
            true, 71, "Republican");
      System.out.println(trump.getName() + " of the " + trump.getPartyMembership()
            + " party " +(trump.canBeElected() ? "can run for President." :
            "cannot run for President."));
      SenatorialCandidate elizabeth = new SenatorialCandidate("Elizabeth II",
            false, 91, "Independent");
      System.out.println(elizabeth.getName() + " of the " +
            elizabeth.getPartyMembership() + " party " +
            (elizabeth.canBeElected() ? "can run for Senate." :
            "cannot run for Senate."));

      trump.updateCampaignFocus(CampaignIssue.IMMIGRATION);
      System.out.println(trump.getName() + "'s campaign focuses on " +
            trump.getCampaignFocus() + ".");
}
```

Which produces the output:

```
Donald Trump of the Republican party can run for President.
Elizabeth II of the Independent party cannot run for Senate.
Donald Trump's campaign focuses on IMMIGRATION.
```

Submission Instructions

Through Blackboard, submit to Vocareum the files **`Electable.java`**, **`PresidentialCandidate.java`**, **`SenatorialCandidate.java`**, and **`CampaignIssue.java`**. Only your last submission will be considered.