Overview

In this practice, you will use a static method, override the `toString` method of the `Object` class in the `Employee` class and in the `Manager` class. You will create an `EmployeeStockPlan` class with a `grantStock` method that uses the `instanceof` operator to determine how much stock to grant based on the employee type.

Assumptions

Tasks

1. Open the `Employee04-01Prac` project in the `practices/practice1` directory.
2. Edit the `Employee` class:

   a. Delete the instance method `printEmployee()`.

   b. Override the `toString()` method from the `Object` class. `Object`'s `toString` method returns a `String`.

      I. Add a `return` statement that returns a string that includes the employee ID, name, Social Security number, and a salary as a formatted string, with each line separated with a newline character (`"\n"`).

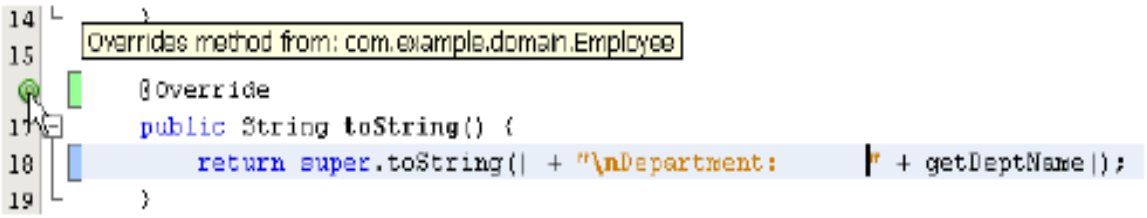
      II. To format the double salary, use the following:

      i.
      ```
      NumberFormat.getCurrencyInstance().format(getSalary())
      ```

      III. Fix any missing import statements.

      IV. Save the class.
3. Override the `toString()` method in the `Manager` class to include the `deptName` field value. Separate this string from the Employee string with a newline character.

   Note the Green circle icon with the "o" in the center beside the method signature in the `Manager` class. This indicates that NetBeans is aware that this method overrides the method from the parent class, Employee. Hold the cursor over the icon to read what this icon represents:

   Click the icon, and NetBeans will open the `Employee` class and position the view to the `toString()` method.
4. (Optional) Override the `toString()` method in the `Director` class as well, to display all the fields of a Director and the available budget.

5. Create a new class called `EmployeeStockPlan` in the package `com.example.business`. This class will include a single method, `grantStock`, which takes an `Employee` object as a parameter and returns an integer number of stock options based on the employee type:

Employee Type	Number of Stock Options
Director	1000
Manager	100
All other Employees	10

a. Add a `grantStock` method that takes an `Employee` object reference as a parameter and returns an integer

b. In the method body, determine what employee type was passed in using the `instanceof` keyword and return the appropriate number of stock options based on that type.

c. Resolve any missing import statements.

d. Save the `EmployeeStockPlan` class.

6. Modify the `EmployeeTest` class:

a. Add a static `printEmployee` method that invokes the toString method of the Employee class.

```
public static void printEmployee(Employee emp) {

        System.out.println(emp);
    }
```

b. Overload the `printEmployee` method to take a second parameter, `EmployeeStockPlan`, and print out the number of stock options that this employee will receive.

i. The new `printEmployee` method should call the first `printEmployee` method and the number of stocks granted to this employee:

```
printEmployee (emp);
System.out.println("Stock Options:    " + esp.grantStock(emp));
```

c. Above the `printEmployee` method calls in the `main` method, create an instance of the `EmployeeStockPlan` and pass that instance to each of the `printEmployee` methods:

```
EmployeeStockPlan esp = new EmployeeStockPlan();
printEmployee(eng, esp);
```

d. Modify the remaining `printEmployee` invocations.

```
printEmployee(adm, esp);
printEmployee(mgr, esp);
printEmployee(dir, esp);
```

h. Modify the code used to display the Managers stock plan after invoking the `raiseSalary` method to

```
printEmployee(mgr, esp);
```

7. Save the `EmployeeTest` class and run the application. You should see output for each employee that includes the number of Stock Options, such as:

```
Employee id:          101
Employee name:        Jane Smith
Employee SSN:  012-34-5678
Employee salary:      $120,345.27
Stock Options:        10
```

8. It would be nice to know what type of employee each employee is. Add the following to your original `printEmployee` method above the print statement that prints the employee data fields:

```
System.out.println("Employee type:        " +
emp.getClass().getSimpleName());
```

This will print out the simple name of the class (`Manager`, `Engineer`, and so on). The output of the first employee record should now look like this:

```
Employee type:        Engineer
Employee id:          101
Employee name:        Jane Smith
Employee SSN:  012-34-5678
Employee salary:      $120,345.27
Stock Options:        10
```