Create a Employee with following data field

1. empId: int
2. empName: String
3. gender: enumerator
4. employeType: enumerator (FULLTYME=10,PARTTIME=20,CONTRACT=30)
5. basicSalary:float
6. hra: float (10% on basic)
7. netSalary:float (basicSalary+hra)

It should contain following methods:

1. getData(): it should take empid,empname
   a. gender as string then convert it to gender enum, and handle exception
   b. same for employee type
   c. basic salary
2. printData()
3. calculate(), it should be overridden from the EmpSalary interface

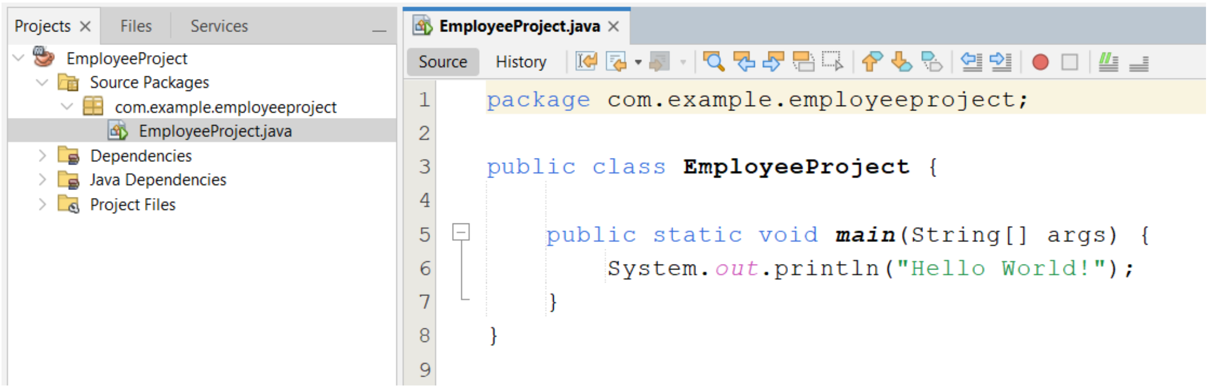
Create an interface called EmpSalary with one method void calculate(), this method should be overridden by Employee class.

Create Employee object in public static void main

Step:

1. open netbeans IDE
2. click on file->new project(categories:java, projectes:java application)->next
3. it will ask the project name, if you give project name as EmployeeProject, then it will create that project and one default class with name EmployeeProject with public static void main.

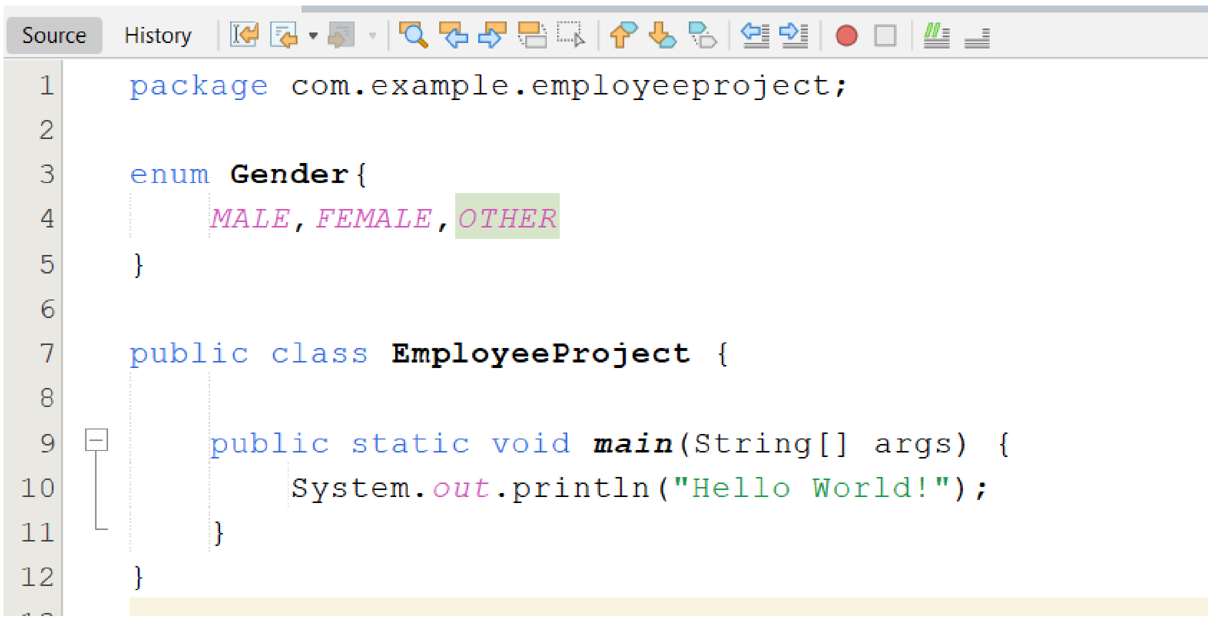
4. Click on finish.



```
package com.example.employeeproject;

public class EmployeeProject {

    public static void main(String[] args) {
        System.out.println("Hello World!");
    }
}
```

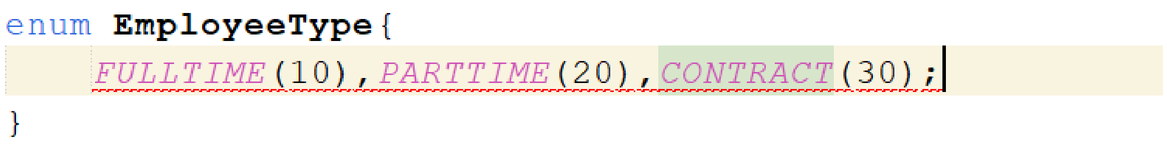
See EmployeeProject.java class is inside com.example.employeeproject folder, we have to write the first line as **package com.example.employeeproject** but this line is automatically generated when we create a class. It should be first line in the file.

5. Now create Gender enumerator in same file, i.e EmployeeProject.java



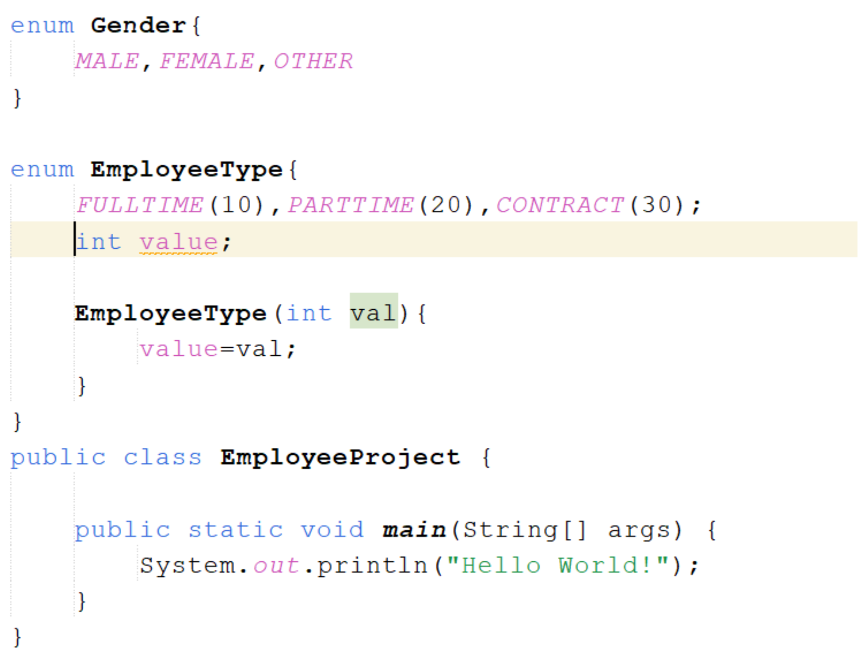
```java
package com.example.employeeproject;

enum Gender{
    MALE, FEMALE, OTHER
}

public class EmployeeProject {

    public static void main(String[] args) {
        System.out.println("Hello World!");
    }
}
```

6. Next create EmployeeType enum in same file. Since each object value is define like FULLTIME it should be 10, create ctor and assign the value



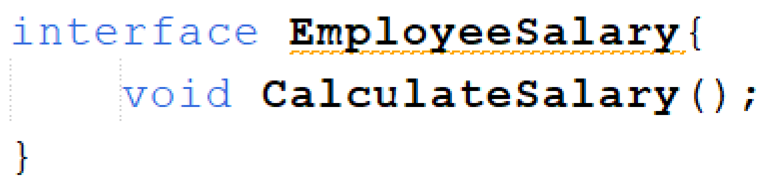
```java
enum EmployeeType{
    FULLTIME(10),PARTTIME(20),CONTRACT(30);
}
```

Now it shows error because there is no ctor. So create one.



```
enum Gender{
    MALE,FEMALE,OTHER
}

enum EmployeeType{
    FULLTIME(10),PARTTIME(20),CONTRACT(30);
    int value;

    EmployeeType(int val){
        value=val;
    }
}
public class EmployeeProject {

    public static void main(String[] args) {
        System.out.println("Hello World!");
    }
}
```

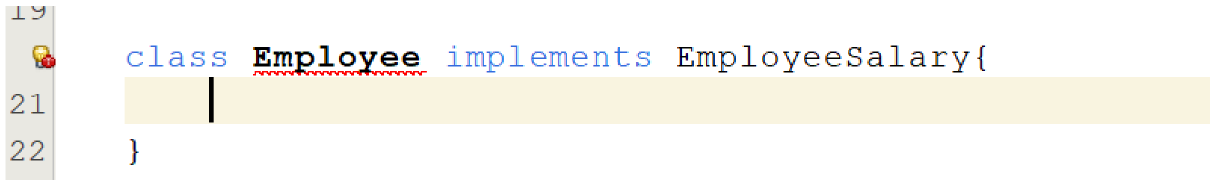
7. Now create interface EmployeeSalary in same file



```
interface EmployeeSalary{
    void CalculateSalary();
}
```

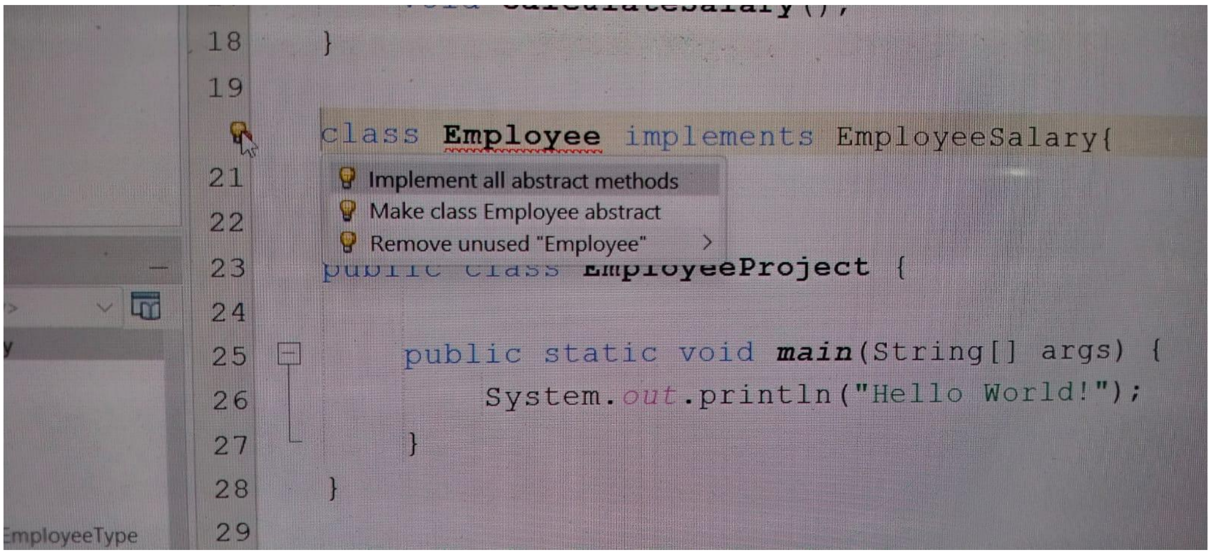
By defualt interface members are public. so internally it will be like public void CalculateSalary();
so while implementing this method you have to write public void CalculateSalary();

8. Now create Employee class with all specification



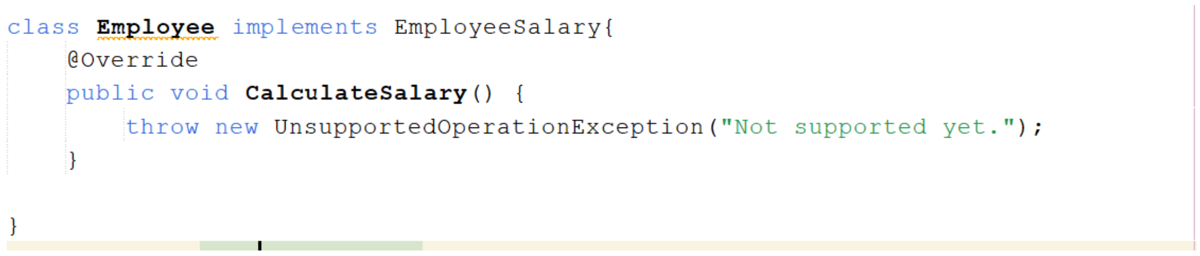
```
class Employee implements EmployeeSalary{

}
```

See if I write like this, it will give error, saying that I have to implement all the methods inside Employeee salary, so click on bulb



```
}

class Employee implements EmployeeSalary{
public class EmployeeProject {

    public static void main(String[] args) {
        System.out.println("Hello World!");
    }
}
```

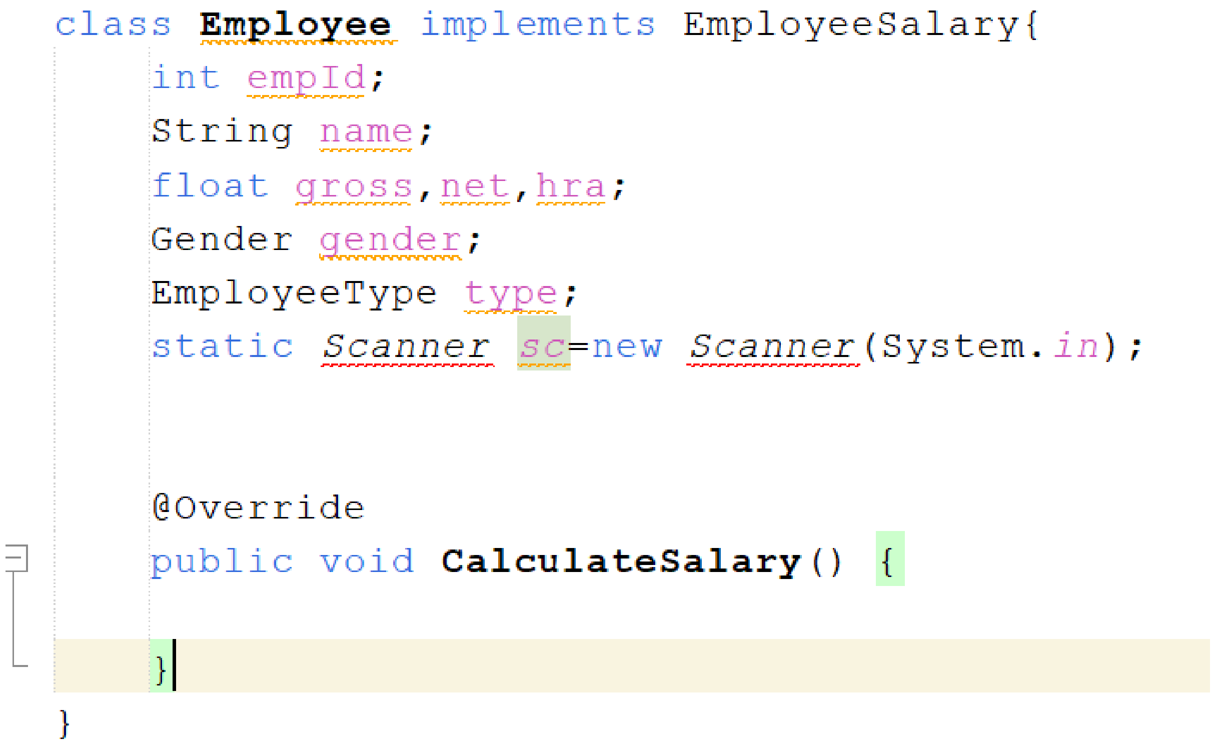
Then click on implement all the methods.



```
class Employee implements EmployeeSalary{
    @Override
    public void CalculateSalary() {
        throw new UnsupportedOperationException("Not supported yet.");
    }

}
```

Now write the code inside that method, and remove that throw error;

Before writing that code, declare all the class member, getData(),display()



```
class Employee implements EmployeeSalary{
    int empId;
    String name;
    float gross,net,hra;
    Gender gender;
    EmployeeType type;
    static Scanner sc=new Scanner(System.in);


    @Override
    public void CalculateSalary() {

    }
}
```

For scanner it is giving me error, so click on that bulb, it will display you import option, import java.util.Scanner

```
class Employee implements EmployeeSalary{
    int empId;
    String name;
    float gross,net,hra;
    Gender gender;
    EmployeeType type;
    static Scanner sc=new Scanner(System.in);

    void getData(){
        try{
            System.out.println("Enter empid, name, gross: ");
            empId=sc.nextInt();
            name=sc.next();
            gross=sc.nextFloat();

            System.out.println("Enter gender: ");
            String g=sc.next(); //it is for gender
            gender=Gender.valueOf(g.toUpperCase()); //if enter gender is not gender object
            //then it will throws an exception. so enclose it inside try catch

            System.out.println("Enter employee type: ");
            type=EmployeeType.valueOf(sc.next().toUpperCase());
        }
        catch(Exception e){
            System.out.println("Invalid gender or type");
        }
    }
    @Override
    public void CalculateSalary() {
```

Now give body to calculateSalary()

```
@Override
public void CalculateSalary() {
    hra=gross*0.1f;
    net=gross+hra;
}
```

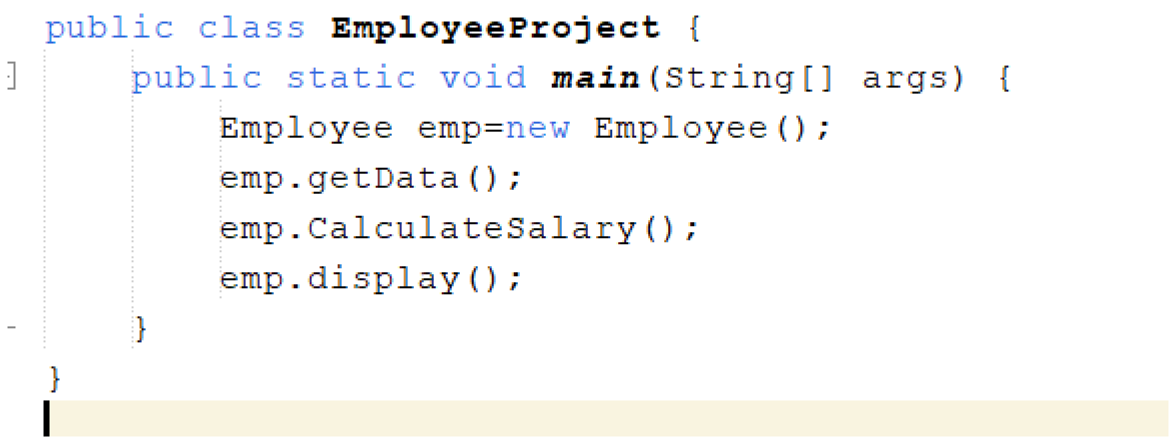
Now give body to display method.

While displaying you can use printf(), which is same as C, or use println()

```
void display(){
    System.out.printf("Name: %s\nGender: %s\nEmpType value: %d",name,gender,type.value);
    //or
    System.out.println("Name: "+name+"\nGender: "+gender+"\nEmpType value: "+type.value);
}
```

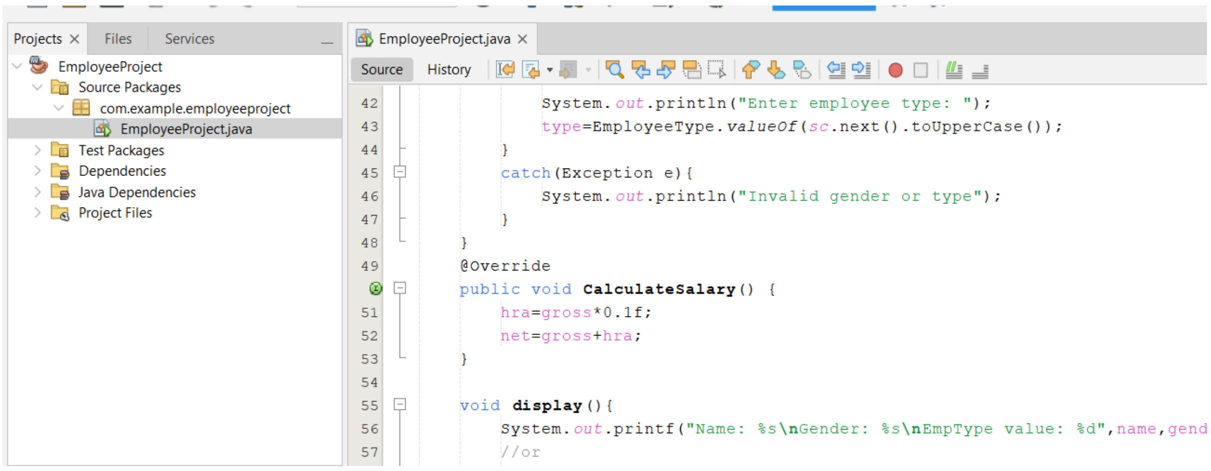
Print other data too

9. Now create employee object in public static void main, and call getData(), calulate(),display(), on that object



```
public class EmployeeProject {
    public static void main(String[] args) {
        Employee emp=new Employee();
        emp.getData();
        emp.CalculateSalary();
        emp.display();
    }
}
```

10. Now compile and run the project, by clicking on play button
11. Final project structure. Everything inside only one file.



```
                System.out.println("Enter employee type: ");
                type=EmployeeType.valueOf(sc.next().toUpperCase());
            }
            catch(Exception e){
                System.out.println("Invalid gender or type");
            }
        }
        @Override
        public void CalculateSalary() {
            hra=gross*0.1f;
            net=gross+hra;
        }

        void display(){
            System.out.printf("Name: %s\nGender: %s\nEmpType value: %d",name,gend
            //or
```

12. Final code.

package com.example.employeeproject;

import java.util.Scanner;

enum Gender{
  MALE,FEMALE,OTHER
}

enum EmployeeType{
  FULLTIME(10),PARTTIME(20),CONTRACT(30);

```
  int value;

  EmployeeType(int val){
    value=val;
  }
}

interface EmployeeSalary{
  void CalculateSalary();
}

class Employee implements EmployeeSalary{
  int empId;
  String name;
  float gross,net,hra;
  Gender gender;
  EmployeeType type;
  static Scanner sc=new Scanner(System.in);

  void getData(){
    try{
      System.out.println("Enter empid, name, gross: ");
      empId=sc.nextInt();
      name=sc.next();
      gross=sc.nextFloat();

      System.out.println("Enter gender: ");
      String g=sc.next(); //it is for gender
      gender=Gender.valueOf(g.toUpperCase()); //if enter gender is not gender
object
      //then it will throws an exception. so enclose it inside try catch

      System.out.println("Enter employee type: ");
      type=EmployeeType.valueOf(sc.next().toUpperCase());
    }
    catch(Exception e){
      System.out.println("Invalid gender or type");
    }
  }
  @Override
  public void CalculateSalary() {
        hra=gross*0.1f;
        net=gross+hra;
    }

    void display(){
        System.out.printf("Name: %s\nGender: %s\nEmpType value:
%d",name,gender,type.value);
        //or
        System.out.println("Name: "+name+"\nGender: "+gender+"\nEmpType value:
"+type.value);
    }
}
public class EmployeeProject {
    public static void main(String[] args) {
        Employee emp=new Employee();
        emp.getData();
        emp.CalculateSalary();
        emp.display();
    }
}
```