

Modificación Patrón Builder

Objetivo

Agregar un departamento al empleado

Resolución

A continuación, se detallarán los pasos realizados para completar el objetivo de la tarea

i. Creación de clase Departamentos “Departments”

```
public class Departments {  
    private String name;  
  
    public Departments() { }  
    public Departments(String name) { this.name = name; }  
  
    public String getName() { return name; }  
    public void setName(String name) { this.name = name; }  
  
    @Override  
    public String toString() { return "Departments{" + "name=" + name + "}"; }  
}
```

ii. Modificación de la clase “Employee”

a. Variable privada para los departamentos “department”

```
private Departments department;
```

b. “Getter and Setter” de la variable privada departamento

```
public Departments getDepartment() {  
    return department;  
}  
  
public void setDepartment(Departments department) {  
    this.department = department;  
}
```

c. Modificación del constructor que acepte el departamento

```
private Employee(String name, int age, String gender, Address adress,  
    List<Phone> phones, List<Contact> contacts, Departments department) {  
    this.name = name;  
    this.age = age;  
    this.gender = gender;  
    this.adress = adress;  
    this.phones = phones;  
    this.contacts = contacts;  
    this.department = department;  
}
```

d. Modificación del método “toString” para que presente la nueva variable departamento

```
@Override  
public String toString() {  
    return "Employee{" + "name=" + name + ", age=" + age + ", gender=" + gender  
        + ", \nadress=" + adress + ", \nphones=" + phones + ", \ncontacts=" + contacts  
        + ", \ndeartment=" + department + '}';  
}
```

- iii. Modificación de la clase “EmployeeBuilder” dentro del archivo de la clase “Employee”
 - a. Creación e inicialización de la clase Departamento “department”


```
private Departments department = new Departments();
```
 - b. Creación del “builder” para agregar el departamento “addDepartments”


```
public EmployeeBuilder addDepartments(String name) {
    this.department.setName(name);
    return this;
}
```
 - c. Modificación del método “build” para que considere el departamento


```
@Override
public Employee build() {
    return new Employee(name, age, gender, adress, phones, contacs, department);
}
```
- iv. Modificación de la clase “BuilderMain”

Se llamo al nuevo “builder” para agregar el departamentos, para agregar el nombre

```
public class BuilderMain {
    public static void main(String[] args) {
        Employee emp = new Employee.EmployeeBuilder()
            .setName("Oscar Javier Blancarte Iturralde")
            .setGender("Male")
            .setAge(29)
            .setAdress("Benito Juarez", "Mexico D.F.", "Mexico", "03400")
            .addContacs("Rene Blancarte", "1122334455", "123", "Casa",
                "Chapultepect No. 123 Col. Militar", "Mexico", "Mexico", "10023")
            .addContacs("Jaime Blancarte", "3344556677", null, "Celular")
            .addPhones("4567890234", null, "Celular")
            .addPhones("7788990099", null, "Casa")
            .addDepartments("Sistemas")
            .build();
        System.out.println(emp);
    }
}
```

Resultado

Finalmente tenemos el siguiente resultado con los cambios realizado:

```
run:
Employee{name=Oscar Javier Blancarte Iturralde, age=29, gender=Male,
adress=Address{address=Benito Juarez, city=Mexico D.F., country=Mexico, cp=03400},
phones=[Phone{phoneNumber=4567890234, ext=null, phoneType=Celular},
Phone{phoneNumber=7788990099, ext=null, phoneType=Casa}],
contacs=[Contact{name=Rene Blancarte, phone=Phone{phoneNumber=1122334455, ext=123,
phoneType=Casa}}, Contact{name=Jaime Blancarte, phone=Phone{phoneNumber=3344556677, ext=null,
phoneType=Celular}}],
department=Departments{name=Sistemas}}
BUILD SUCCESSFUL (total time: 0 seconds)
```

