Job Sheet 04 ‑ Relation Class

# Competence​

After to go through main discussion This, student capable:

* 1. Understand draft relation class;
  2. Implementing association relationship to in program.

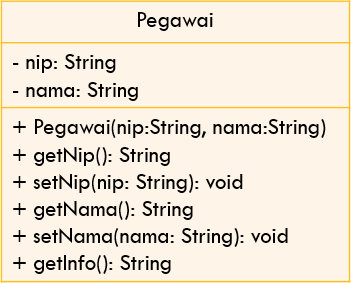
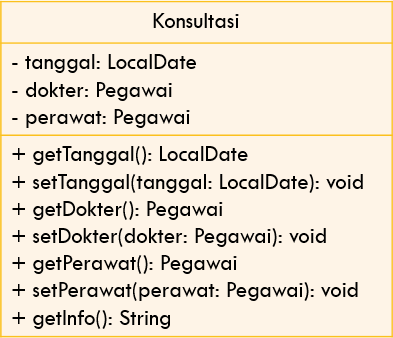
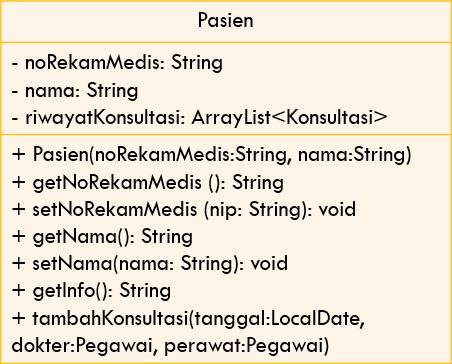
# Introduction

In more complex cases, more than one will be found in a system. *class* Which each other own relatedness between *class* One with Which other. On previous experiments, the majority of cases that have been worked on only focus on One *class* just. On job sheet This will done test Which involving a number of *class* Which each other related.

# Practicum

On practical work This will developed a system information House Sick Which save data history patient consultation.

Take note *class* diagramfollowing:



1

0..\*

0..\*

0..\*

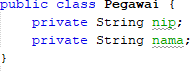
merawat

memeriksa

1

1

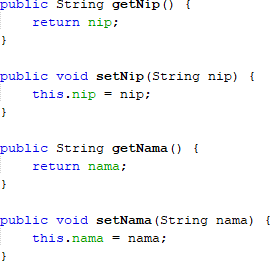
1. Make it folder new with Name Hospital
2. Make it class Employee. Add attribute nip And Name on class Employee with access modifier private



1. Make it *constructor* For class Employee with nip parameters And Name.

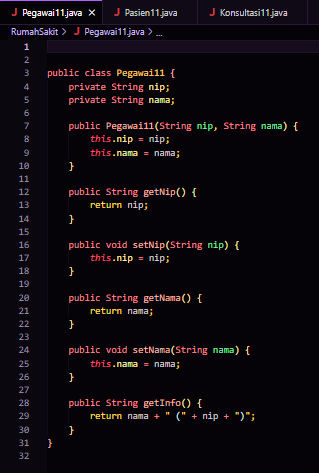


1. Implement **setter** And **getter** For class Employee.

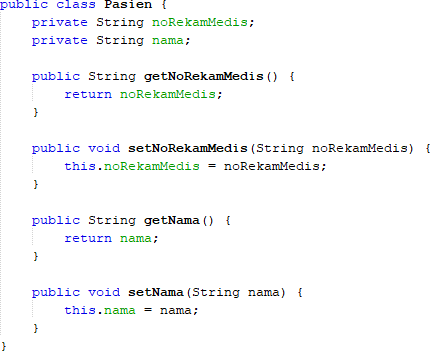


1. Implement *method* getInfo()as following:

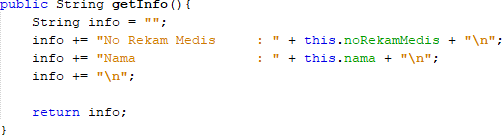


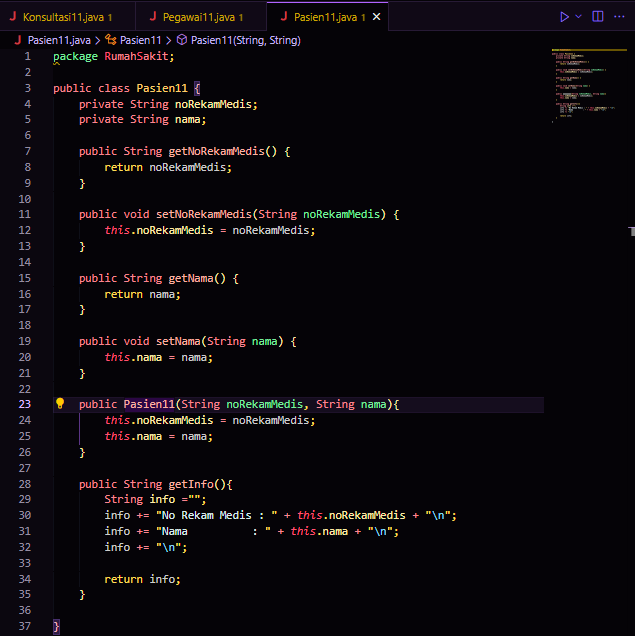


1. Furthermore make it class Patient Then add attribute noMedicalRecord And name in the Patient class with the access level modifier private. Also provide setters and getter for second attribute the.
2. Make it constructor For class Patient with noRekorMedic parameter And Name

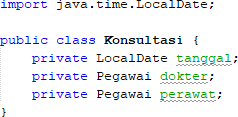


1. Implement *method* getInfo()as following:

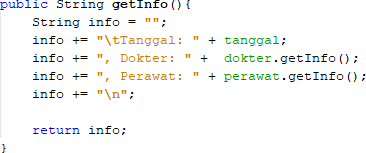
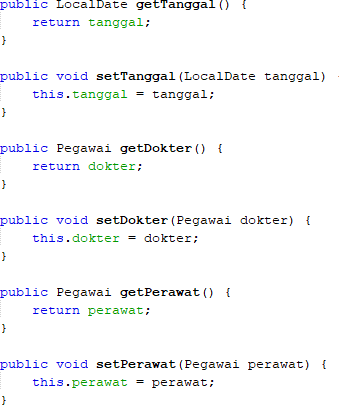




1. This system will store data on every consultation that the patient has. Patients can conduct consultations more than once. Therefore, the consultation data will be stored in ArrayList form from objects Which type Consultation.
2. Make it class with Name Consultation with attribute date type LocalDate, doctor is of type Employee, and nurse is of type Employee. Set access level modifier private For all over attribute. Do import java.time.LocalDate so that can to declare attribute date type LocalDate.
3. Provide setter And getter For each attribute on class Consultation

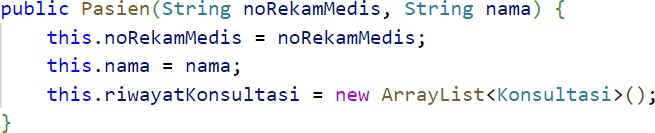


1. Implement method getInfo() as following:

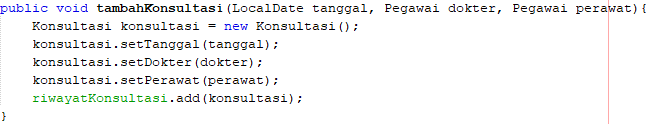




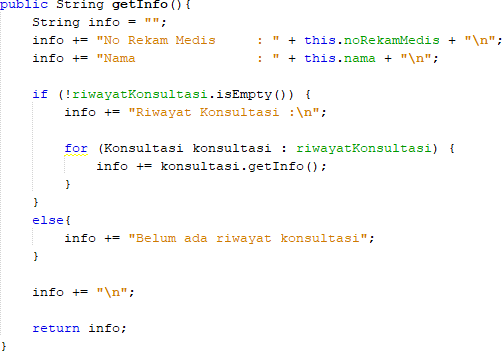
1. For keep data history consultation patient, so add attribute historyConsultation on class Patient with type arrayList<Consultation>. Attribute This will store a series of objects of type Consultation. Import java.util.ArrayList so that can to declare attribute type ArrayList of object.
2. Make it constructor parameterized For class Patient. Initiation mark attribute noRekamMedis and name based on name attribute. Instantiate/create new ArrayList For attribute historyConsultation;

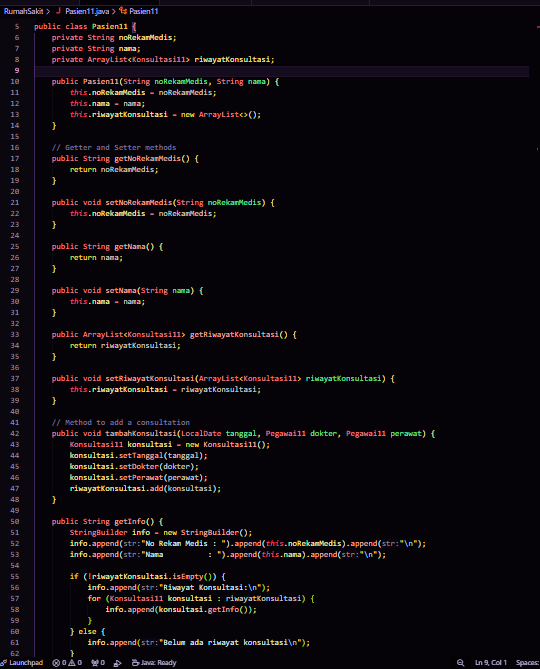


1. Do import java.time.LocalDate so that can to declare attribute date type LocalDate on class Patient. Furthermore, implement method addConsultation() as follows:

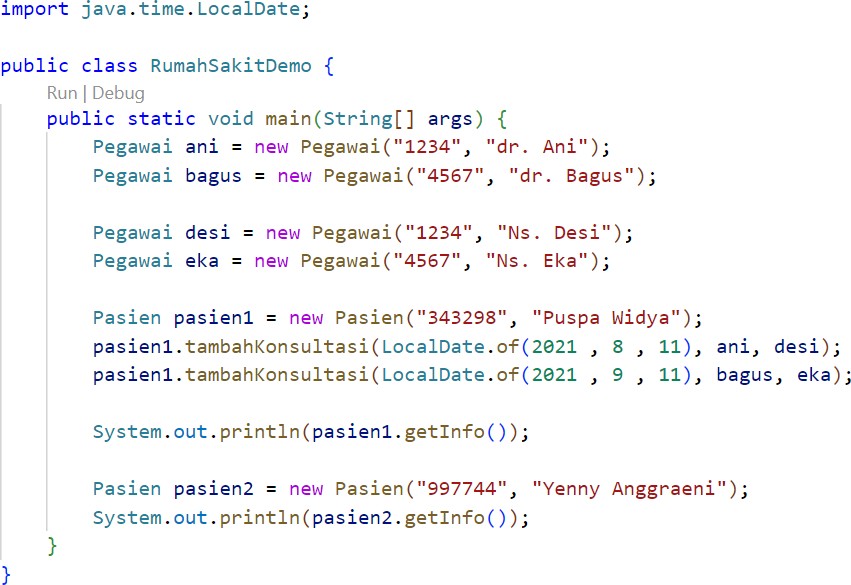


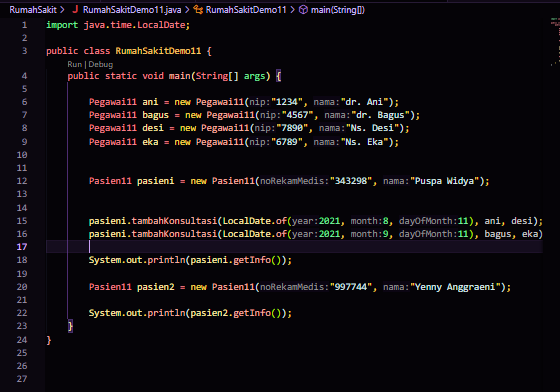
1. Modification method getInfo() For return info patient And list consultation ever done



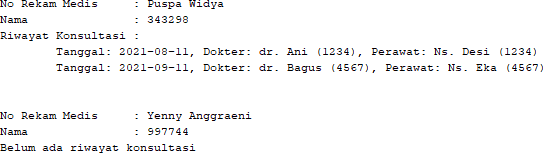


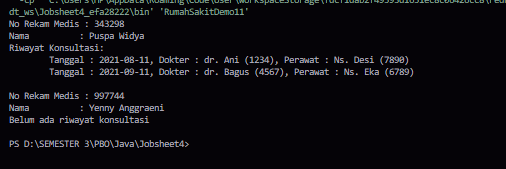
1. Do import java.time.LocalDate so that can to declare attribute date type LocalDate on class HospitalDemo. Test program Which Already made with make objects on class HospitalDemo. Instantiation object new type Employee with Name ani use constructor Employee(String nip, String Name) with mark argument nip “1234” And Name “dr. Ani”. Continue instantiation object as follows:





1. *Compile* Then *run* HospitalDemo and obtained results like following:





# Question

Based on test 1, answer it questions Which related:

1. In in *class* Employee, Patient, And Consultation, there is method *setter* And *getter*

For each its attributes. Whether the use *of method setter* And *getter* the ?

* **Employee Class:**

**Getter: To retrieve the nip value and employee name.**

**Setter: To change the employee's NIP and name values, and can add validity checks.**

* **Patient Class:**

**Getter: To retrieve the Medical Record number and patient name values.**

**Setter: To change the Medical Record number and patient name values, and can check the validity of the data.**

* **Consultation Class:**

**Getter: To retrieve the values of date, doctor, and nurse consultation.**

**Setter: To change the values of the date, doctor, and nurse consultation, and can adjust the data if necessary.**

1. In in *class* Consultation No in a way explicit there is constructor with parameter. Is this means class Consultation No own constructor?

**Yes, the Consulting class still has a constructor, even though it is not explicitly defined. Java automatically provides a default constructor with no parameters.**

1. Take note *class* Consultation, attribute where just Which type *object* ?

**A attribute of type object :**

* **doctor**
* **nurse**

1. Take note *class* Consultation, on line which one Which show that *class*

Consultation own relation with *class* Employee?

**In the Consultation class, the relationship with the Employee class is seen in the line :**

* + **private Doctor's staff;**
  + **private Nursing staff;**
  + **This line shows that Consultation uses the Employee class for the doctor and nurse attributes, so there is a relationship between Consultation and Employee.**

1. Take note on *class* Patient, What Which done by code consultation.getInfo() ?

**In the Patient class , the code consult.getInfo() is used to returns detailed information about the consultation object , such as date, doctor, and nurse, in string form.**

1. On method getInfo() in class Patient, there is line code:

if (!consultationhistory.isEmpty())

Whether Which done by line the?

**The code line if (!rihiwatKonsultasi.isEmpty()) checks whether the list of sejarahKonsultasi contains data.**

**If There Is Data : The code inside the if block will be executed , displaying the consultation history information.**

**If Empty : The code inside the else block will be executed** , **stating that there is no consultation history yet** .

1. In the Patient class constructor, there is a line of code: this.historyConsultation = new ArrayList<>();

Whether Which done by line the? Whether Which happen If line the removed?

**The code line this.rihwatKonsultasi = new ArrayList<>(); in the Patient class constructor does :**

**Initialize List : Create and assign an empty list for ConsultationHistory .**

**If removed:**

**UninitializedConsultationHistory : ConsultationHistory will be null , causing an error when trying to access it.**

# Task

Implement studies the case that has made on PBO duties Theory to in program