**[CMP368] Practice: Object-oriented Programming**

**LAB 1: Getting Started With Java & IntelliJ IDEA**

**Course:** Object – Oriented Programing

**Class:** 23DTHQA1

**Full Name:** Trần Duy Thành

**ID:** 2380602052

**Question** :

100 times. • Modify code to print the configured arguments:

Hello Nguyen Thanh Tung 100 times.

Hint: – To access the arguments (it is an array): use args[i]

where i is an integer, e.g., args[0], args[1], args[2]

public class Main {  
 public static void main(String[] args) {  
  
 String message = args[0] + " " + args[1] + " " + args[2];  
  
 for (int i = 0; i < 100; i++) {  
 System.*out*.println(message);  
 }  
 }  
}

**Question 1 (page9):** Click the green triangle to run the main method:

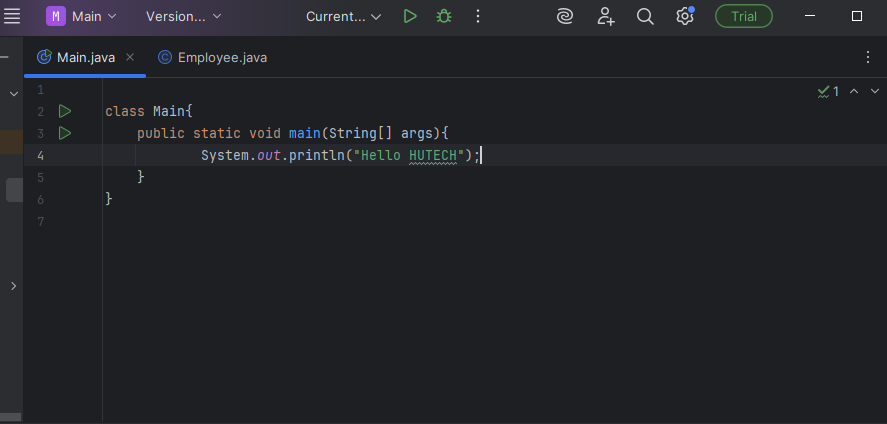
– How many green triangles do you see? What is the difference?

**Answer:** There are 3 green triangle in the software (include the all from the file to the each method/class). The difference of three is :

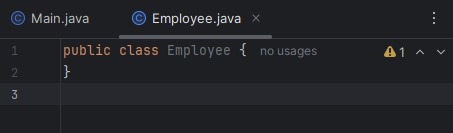
+ 1 is run file IDE (all classes inside the file).

+ the other (next to the class declaration) is run for class.

+ And the last one is run for method, it’s run only that method.



**Question 2 (page 12):** Look into the code of the Employee class, do you see a warning indicated by a yellow triangle? What is that?



**Answer:** The warning in Employee.java indicates that the class Employee is declared but never used in the project. This means no code is creating or referencing an Employee object yet. It is just a definition without usage.

**Question 3(Page 13):**

In Employee class, write code to add a method:

– Name: getSalary

– Parameter:

• Name: numDays

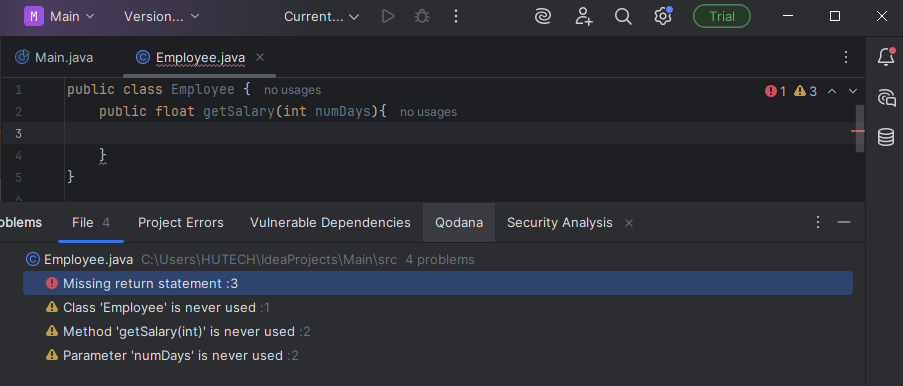
• Type: int

– Return type: float

– Body: empty

• Notice the error, explain it.

**Answer:** Java oblige user to return any value in the method. If that method have no value to return, have a error in the code because method wrote that it was return type “float”.

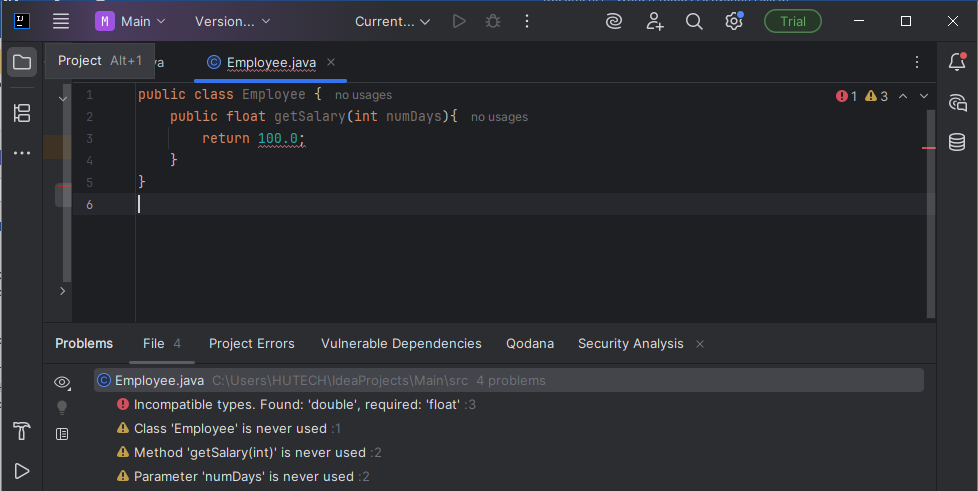


• In the method body, write:return 100.0;

**– Notice the error, explain it.**

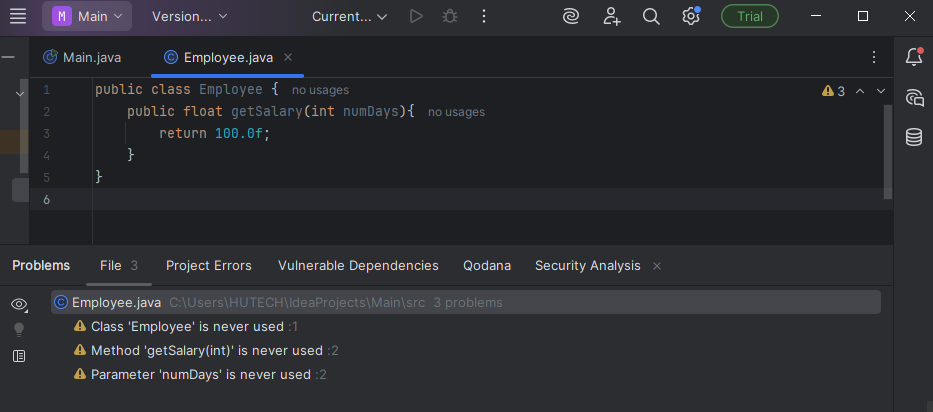
**Answer: double literal** by default in Java.

It’s cannot directly assign a double to a float without casting and the compile error: “Incompatible types. Found: 'double', required: 'float' ”.



**– Modify the body to fix the error.**

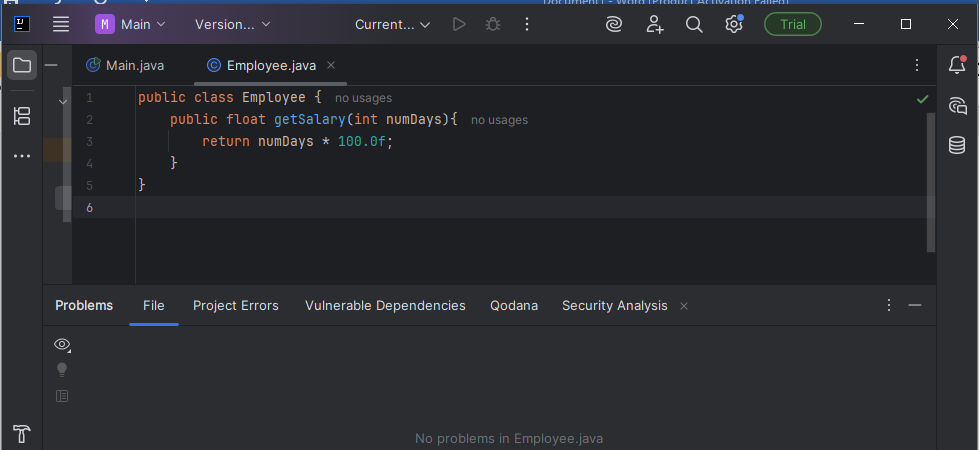
**Answer:**

****

• Notice the warning about numDays. Modify the body to fix that warning.

**Answer:**

To fix the warning, use numDays in the logic.

****

**Question 4(Page 14):**

**Add 2 more methods as below:**

int getSalary(int numMonths) {

return 100;

}

float getSalary(int numMonths, int numDays) {

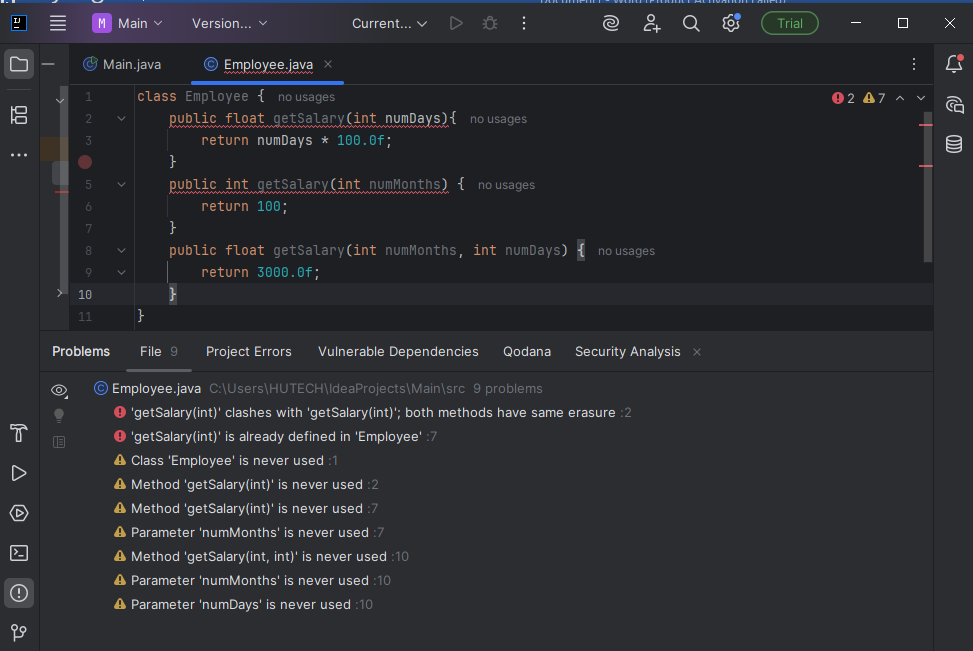
return 3000.0f;

}

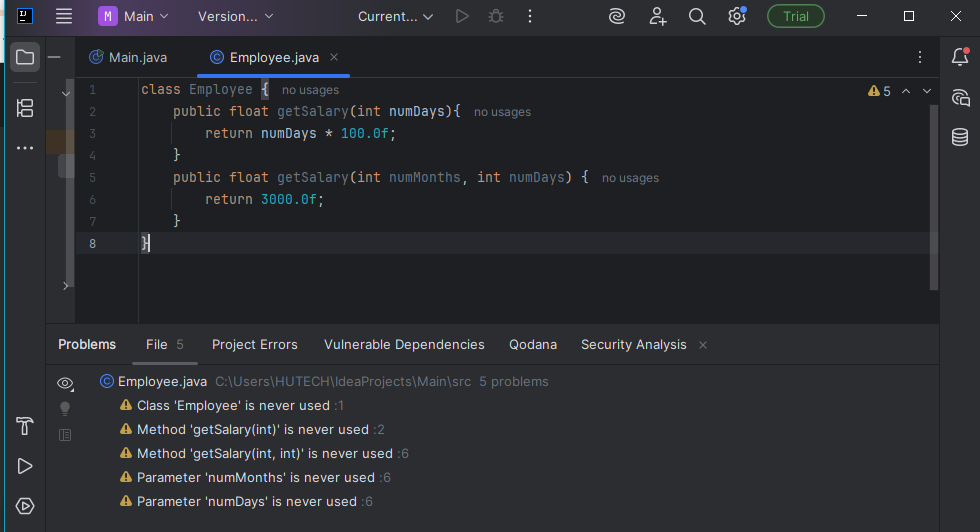
**• Notice the error, explain it. You may need to review the lecture about Method Overloading.  
Answer:**

The methods getSalary(int numDays) and getSalary(int numMonths) have the same signature because overloading is based only on the number and types of parameters, not on their names or return type.

With**method overloading**, multiple methods can have the same name with different parameters. It’s mean that in this code can’t have two methods have same parameters

****

**Fix-ed :**

****

**Question 5(page 15):**

**• Implement the two getSalary methods with respect to salaryPerHour.**

**Assume that 1 month always = 30 days.**

class Employee {  
 public float getSalary(int numDays){  
 return numDays \*8\* salaryPerHour ;  
 }  
 public float getSalary(int numMonths, int numDays) {  
 int totalDays = numMonths \* 30;  
 return getSalary(totalDays);  
 }  
 private float salaryPerHour = 0.0f;  
}

**Question 6(Page16):**

Write code to add 2 methods getSalaryPerHour and setSalaryPerHour

• Constraint: salaryPerHour cannot be negative.

– If the parameter’s value is negative, the setSalaryPerHour method does nothing.

• Please implement the body of these 2 methods.

private float salaryPerHour = 0.0f;  
 public void setSalaryPerHour(float salaryPerHour) {  
 if (salaryPerHour >= 0) {  
 this.salaryPerHour = salaryPerHour;  
 } else {  
 this.salaryPerHour = 0.0f;  
 }  
 }  
 public float getSalaryPerHour() {  
 return salaryPerHour;  
 }

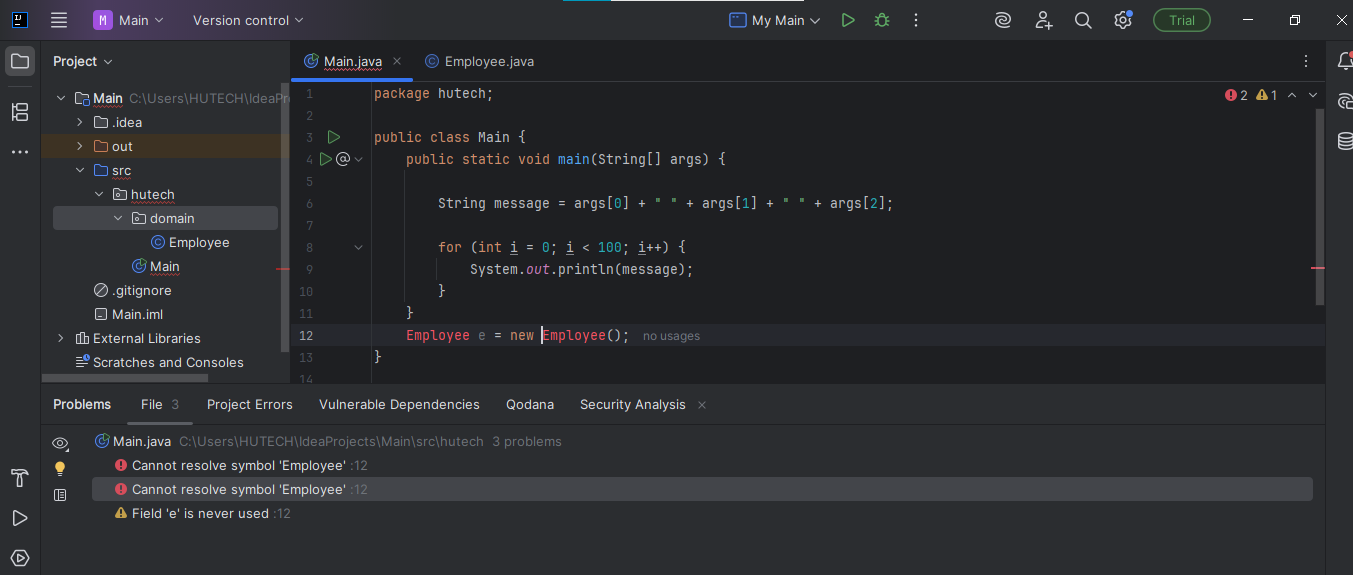
**Question 7(Page 19):**

In the main method of the Main class:

– Create a new instance (object) of the Employee class.

– Notice how IDEA assists you to import the desired class in the corresponding package.

– Notice the warning about the newly created instance.

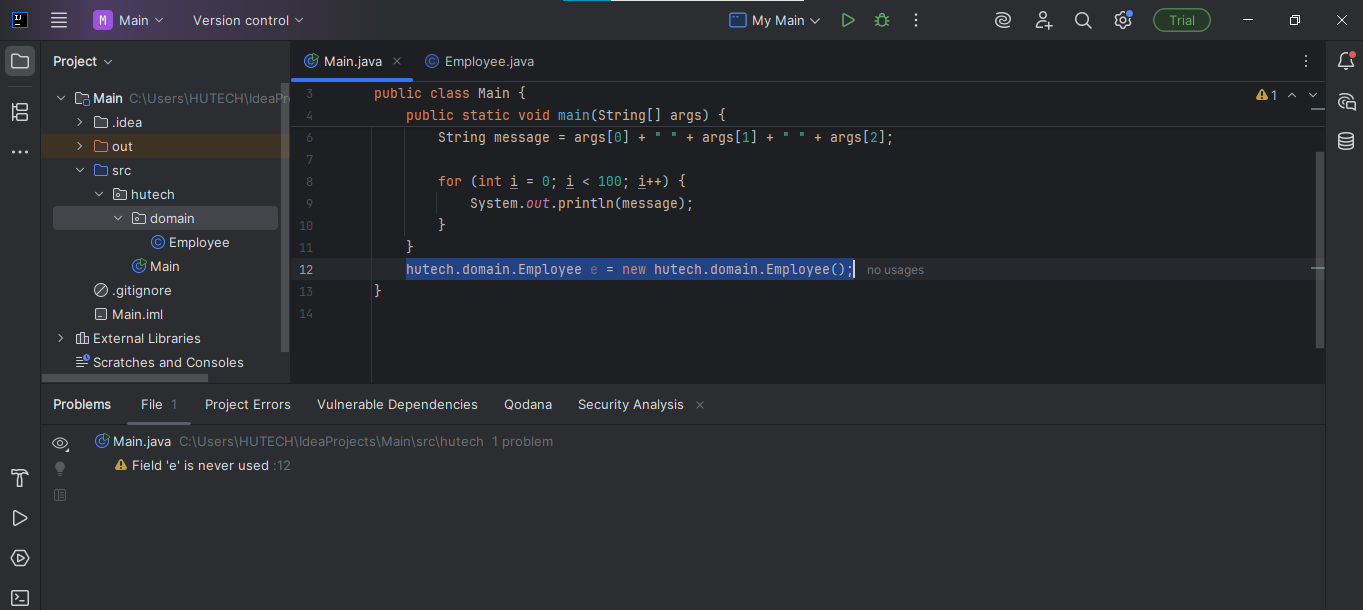
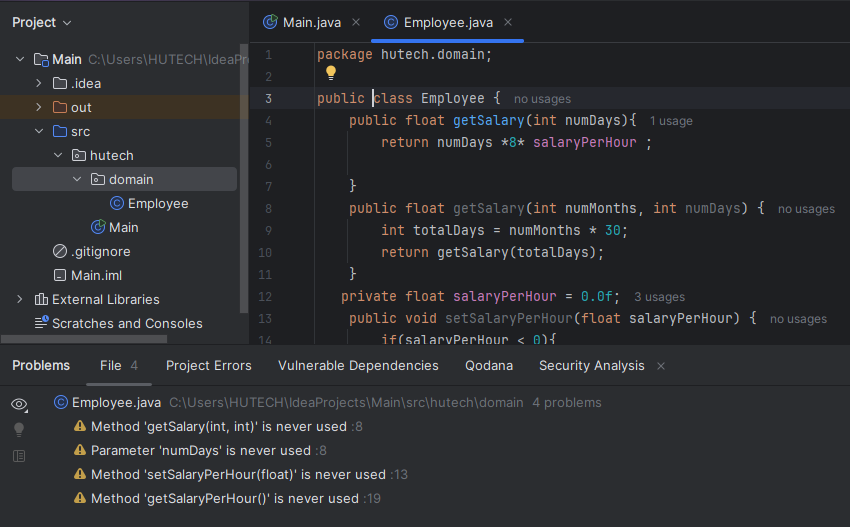
**Answer : **

• Remove the import statement:

– Notice the error.

– Write code to fix the error (without import, of course).

**Answer :** It’s have to public the class Employee

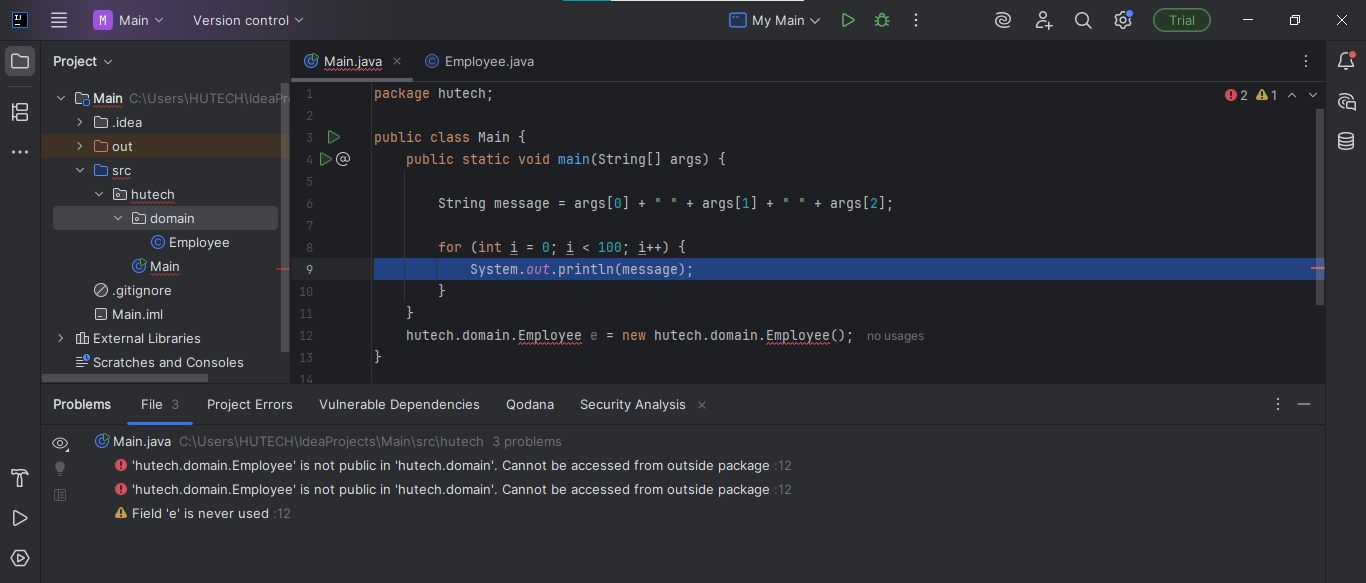
**  
**

**Question 8(page 20):**

• Make the Employee class available only in its package.

• Back to the Main class, notice the error, explain it.

**Answer:** This is the error of the program when I make the Employee class is private. Because the error happens because the Employee class is declared without the public keyword, which means it has default (package-private) access. With this access modifier, the class can only be accessed inside its own package (domain). Since the Main class is in a different package (hutech), it cannot access Employee

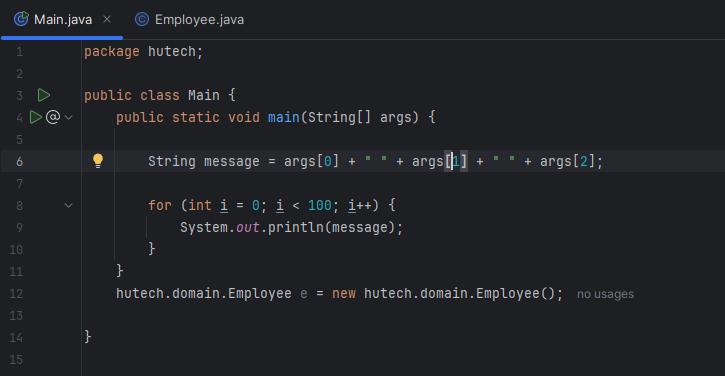
****

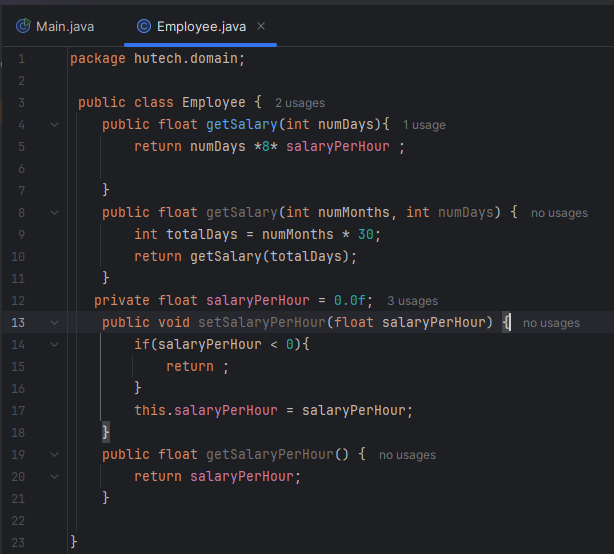
• Make the Employee:

-Make all attributes **private**

-Make all method is **public**

**Answer:**

****



**Question 9(Page21):**

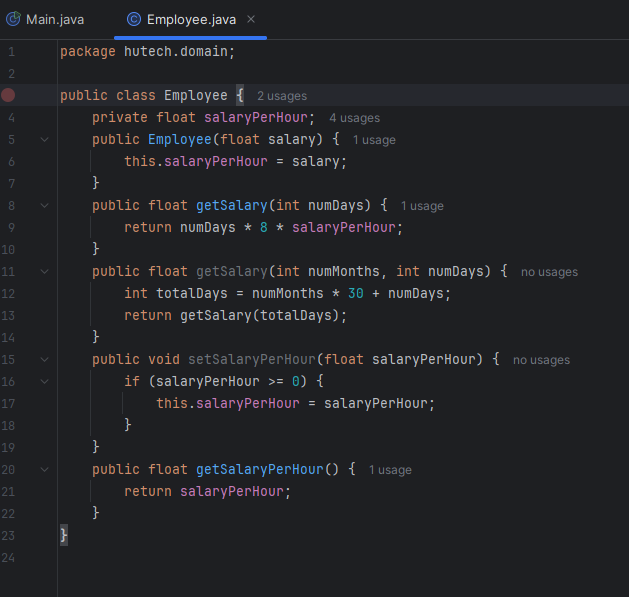
**In the main method of the Main class:**

– Use the object which was already created and call its getSalaryPerHour method, then print the returned value.

– Run and notice the printed value.

**• In the Employee class:**

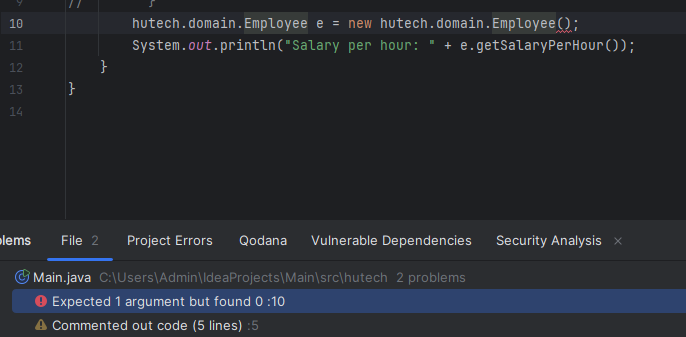
– Write a constructor that receives and stores the value of salaryPerHour



**• Back to the main method of the Main class :**

– Notice the error, explain it.

**Answer :** the compiler cannot find a matching constructor. The error message is shown because the constructor requires a float argument but no argument was provided

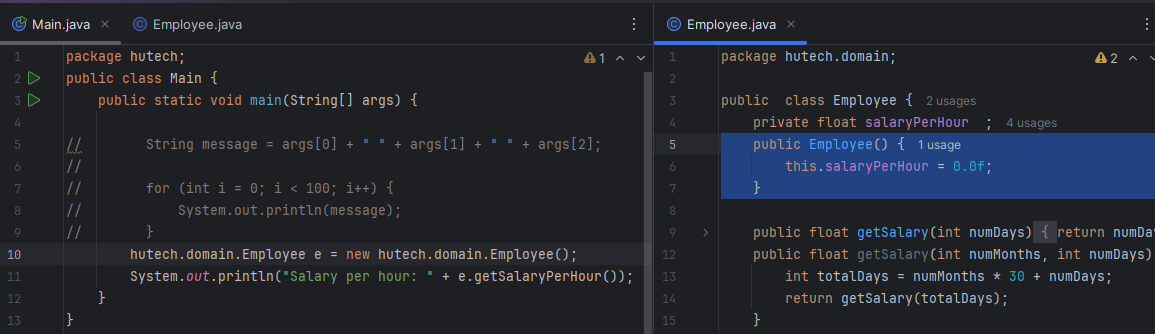
****

**Question 10 (page 22) :**

Back to the Employee class:

– Modify code to fix the error above.

**Answer :**

****

**• Back to the main method of the Main class:**

– Create another instance of Employee with salaryPerHour = 7.0f passed to the constructor.

– Call its getSalaryPerHour method and print the returned value.

– Notice that the value passed into the constructor overrides the initial value.

**Answer :**

****

**• As you can see, locating & switching between classes is a common task in programming.**

**Question 11(Page 23) :**

In Employee class, write a static method:

static int sum(int a, int b) {

return a + b; }

• In the getSalary methods, try to call the sum method. Is that possible?

**Answer : yes. We can call the sum method in the getSalary.Here is the code :**static int sum(int a, int b){  
 return a+b;  
}  
  
public float getSalary(int numDays) {  
 int totalsHours = *sum*(numDays,0)\*8;  
 return numDays \* 8 \* salaryPerHour;  
  
}

• In the sum method, try to call the getSalary methods and the salaryPerHour attribute. Is that possible?

**Answer :** No. A static method cannot directly access instance variables or instance methods. To do so, you must create a new Employee object