Philip Pesic

Week 16

May 7 2023

Week 16 Prog 4

Write a class with overloaded methods:

* Figure out a useful example and code it.

```
package Prog4;
class Employee {
  private String department;
  public Employee(String name, int age, String department) {
      this.age = age;
      this.department = department;
  public Employee(String name, int age, String department, double salary) {
      this.department = department;
  public double getSalary() {
       Employee emp1 = new Employee("Jeremy", 20, "Sales");
```

Philip Pesic

Week 16

May 7 2023

Week 16 Prog 4

```
Employee emp2 = new Employee("Ben", 17, "Engineer", 90000);
                           System.out.println(emp1.getSalary());
                            System.out.println(emp2.getSalary());
                                                                                Prog4 > J prog4.java > 4 prog4 > 9 main(String[])

1 package Prog4;
 ∨ PROGRAMS
       J prog1.java U 2

∨ Prog2 4

J prog2.java U 5
     J prog2.java ∪
∨ Prog3
                                                                                         private String name;
private int age;
private String department;
private double salary;
                                                                                      9
10 public Employee(String name, int age, String department) {
11 this.name = name:
                                                                                                            tmptoyee(String name, int a
    this.name = name;
    this.age = age;
    this.department = department;
}
                                                                                                              public Employee(String name, int :
    this.name = name;
    this.age = age;
    this.department = department;
    this.salary = salary;
}
                                                                                                                  public Employee(String name, int age, String department, double salary) {
                                                                                     17 this.mi this.am this.di this.di this.di this.di 20 this.di 21 22 public doul return } 23 class prog4 { 30 Run | Debug | Run | R
                                                                                                                   public double getSalary() {
    return salary;
}
                                                                                                                       Run|Debug
public static void main(String[] args) [
Employee emp1 = new Employee(name: "Ben"), age: 10, department: "Sales");
Employee emp2 = new Employee(name: "Ben", age: 17, department: "Engineer", salary: 90000);
                                                                                                                             System.out.println(emp1.getSalary());
System.out.println(emp2.getSalary());
                                                                                                                                  System.out.println(x: "Philip Pesic 5/7/23");
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            & Run: prog4 + ∨ □ 葡 ^ ×
                                                                                      0.0
90000.0
Philip Pesic 5/7/23
→ programs git:(main) x
> OUTLINE
 > TIMELINE
```

I practiced overloading my own methods