# OODP workshop 10

# Student Class

1. Think about a class named Employee in payroll system of an organisation.

Ans: We will think of a payroll system which have an Employee class with Attribute member variables and Behaviour Methods. We will put this code in Employee.java file.

1. What are the different attributes that an Employee class can have?

Ans: **Attributes of the Employee class:**

1. name: String
2. employeeId: int
3. dateOfBirth: LocalDate
4. address: String
5. phoneNumber: String
6. email: String
7. department: String
8. designation: String
9. dateOfJoining: LocalDate
10. salary: double
11. paymentMethod: PaymentMethod
12. bankDetails: BankDetails
13. leaveBalance: int
14. employeeType: EmployeeType
15. taxRate: double

1. What are the methods that an Employee class can have?

Ans:

**Methods of the Employee class:**

1. Constructors: Default, parameterized with essential attributes, parameterized with most attributes.
2. Getters and Setters for each attribute.
3. calculateSalary()
4. applyLeave(int daysToApply)
5. updatePersonalInfo(String address, String phoneNumber, String email)
6. changeDepartment(String newDepartment)
7. promoteEmployee(String newDesignation, double newSalary)
8. terminateEmployee(LocalDate terminationDate)
9. generatePayslip()
10. calculateTaxDeduction()
11. updateBankDetails(BankDetails newBankDetails)
12. requestLeave(int daysToRequest)
13. printEmployeeDetails()
14. calculateNetPay()
15. How many constructors this class can have?

* Ans: There are three constructors: default, parameterized with essential attributes, parameterized with most attributes.

1. How many getters and setters this class can have?

Ans: Each attribute has a getter and a setter. With 15 attributes, there will be 15 getters and 15 setters, making a total of 30 getter and setter methods.

1. What a toString() method will display in this class?

Ans:

@Override

public String toString() {

return "Employee{" +

"name='" + name + '\'' +

", employeeId=" + employeeId +

", dateOfBirth=" + dateOfBirth +

", address='" + address + '\'' +

", phoneNumber='" + phoneNumber + '\'' +

", email='" + email + '\'' +

", department='" + department + '\'' +

", designation='" + designation + '\'' +

", dateOfJoining=" + dateOfJoining +

", salary=" + salary +

", paymentMethod=" + paymentMethod +

", bankDetails=" + bankDetails +

", leaveBalance=" + leaveBalance +

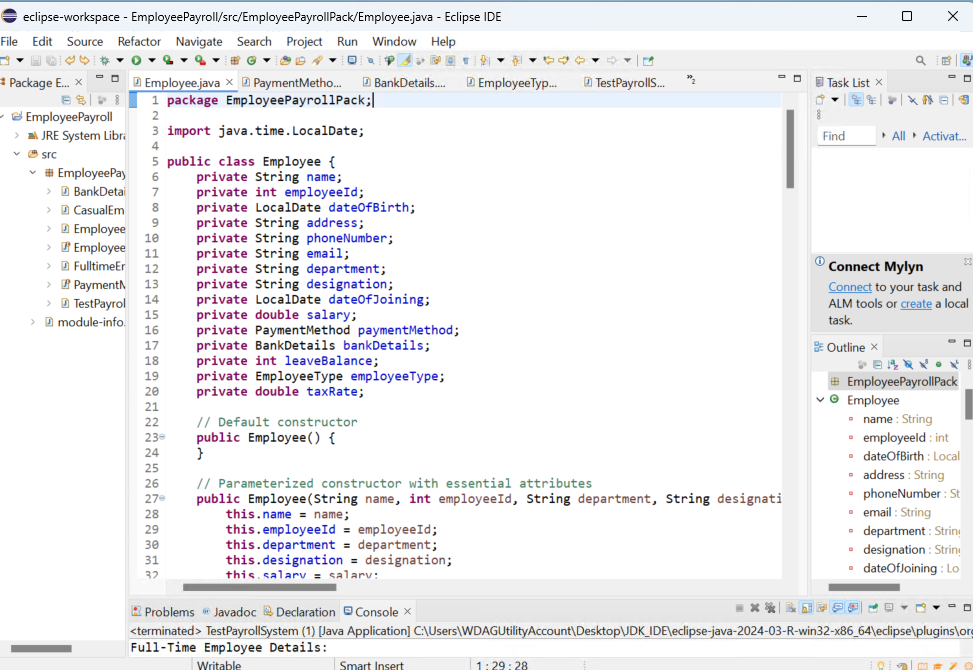
", employeeType=" + employeeType +

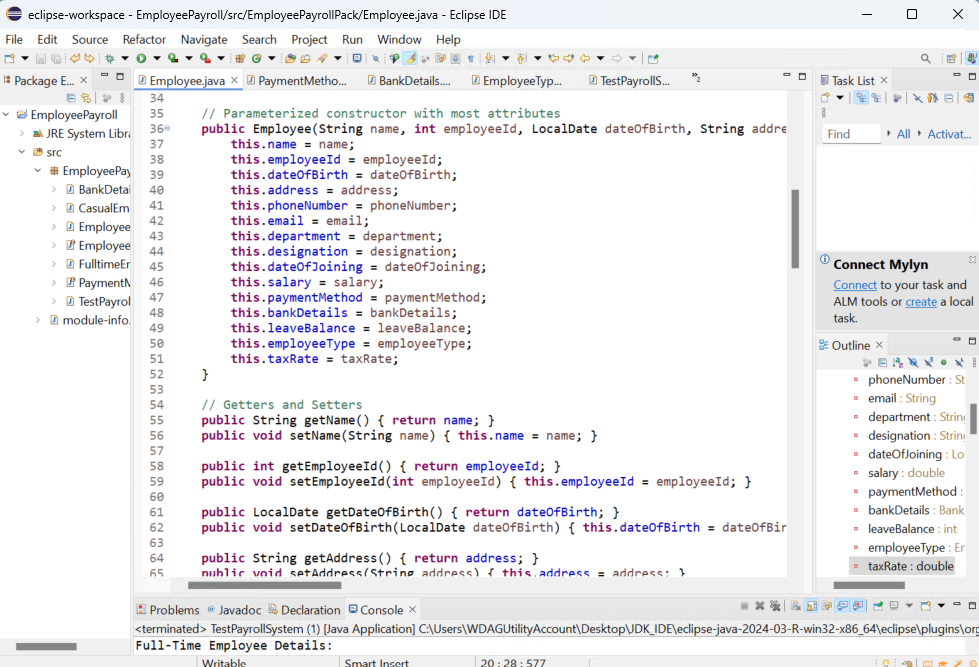
", taxRate=" + taxRate +

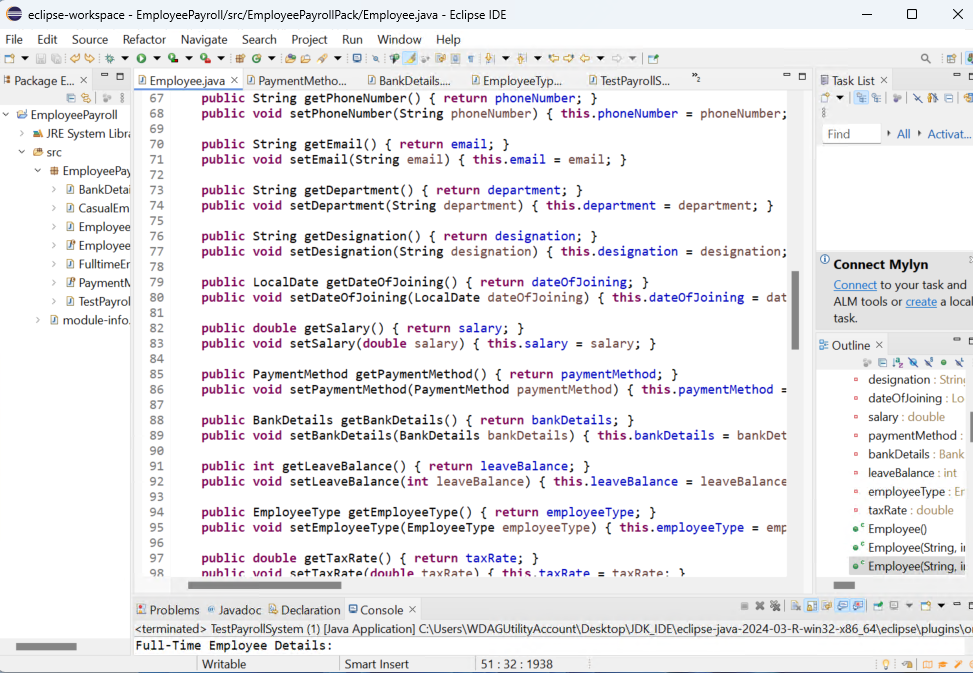
'}';

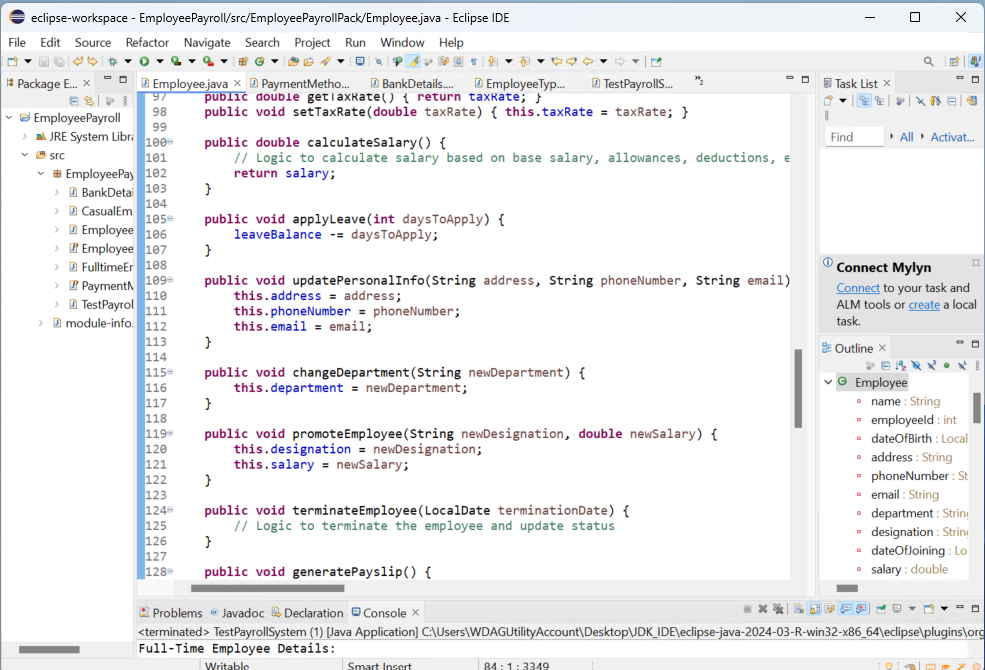
}

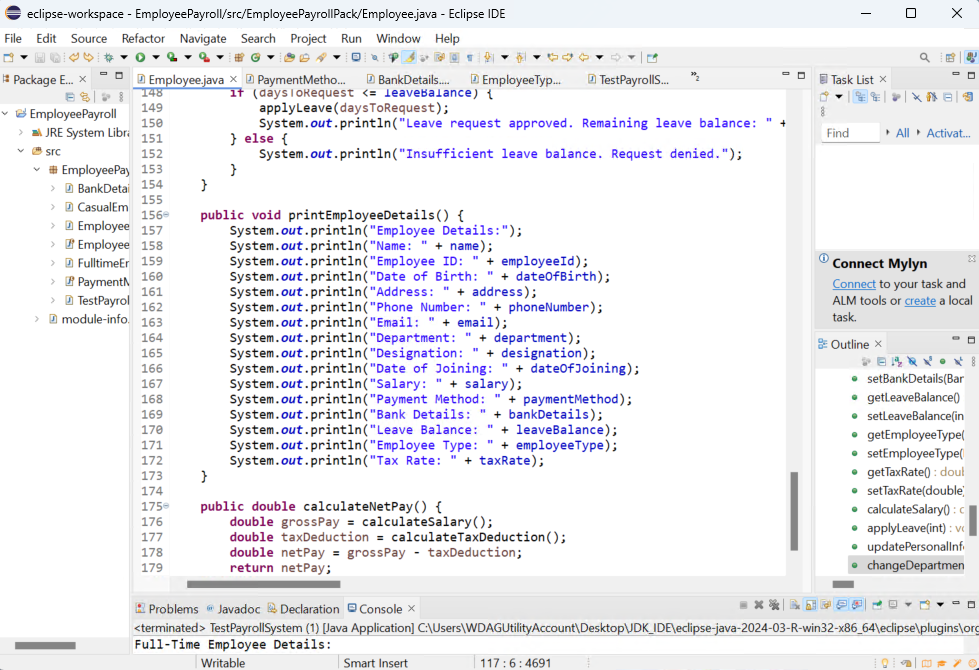
Open Eclipse, and develop this class on the basis of information you have written in answers above.











# FulltimeEmployee Class

Consider FulltimeEmployee class as a sub class of Employee

1. What is the additional attribute that a FulltimeEmplloyee class can have?

Ans: benefits: String

1. Discuss the different types of constructor that this class can have and how will you use “Super” keyword?

Ans: **Constructors and usage of super:**

1. Default constructor.
2. Constructor with essential attributes.
3. Constructor with all attributes including the additional one.
4. Write getter and setter for additional attributes and override the toString() method of Employee class which will display the all details of that particular FulltimeEmployee.

Ans: // Getter and Setter for bonus

public double getBonus() {

return bonus;

}

public void setBonus(double bonus) {

this.bonus = bonus;

}

@Override

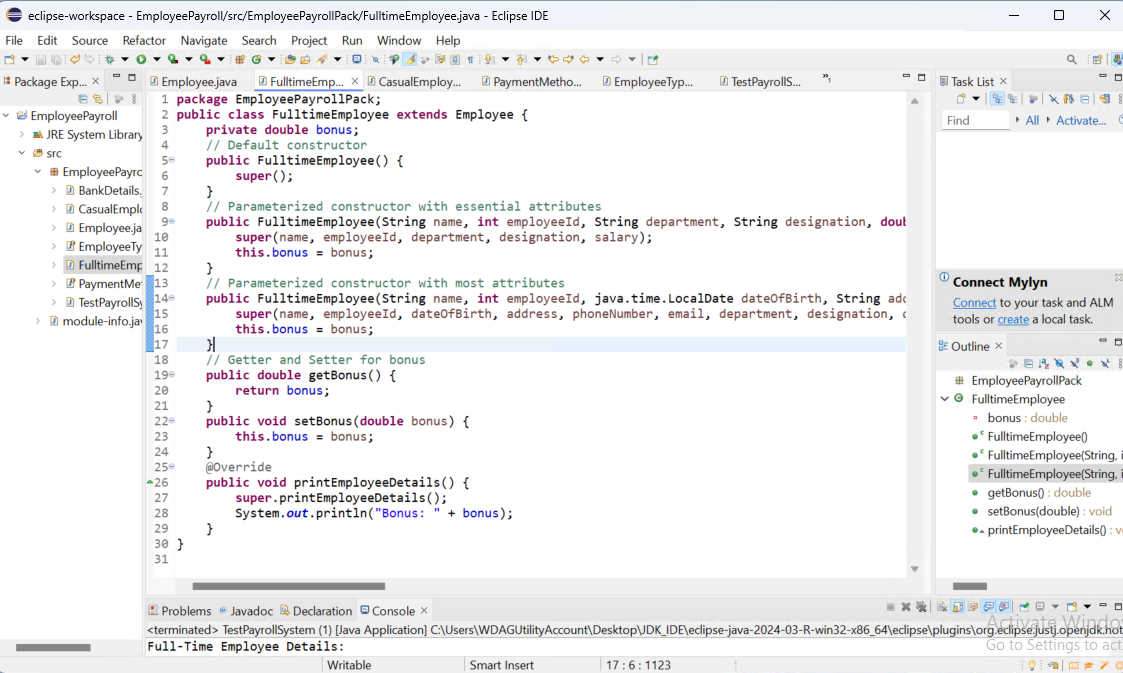
public void printEmployeeDetails() {

super.printEmployeeDetails();

System.out.println("Bonus: " + bonus);

}

}



# CasualEmployee Class

Consider CasualEmployee class as a sub class of Employee

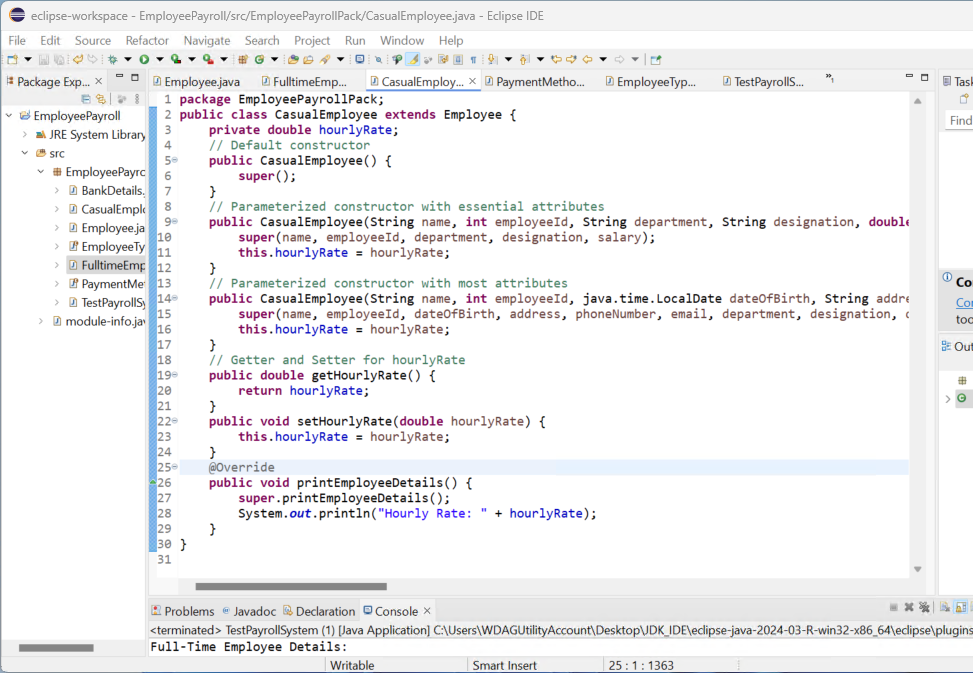
1. What is the additional attribute that a CasualEmployee class can have?

Ans: hourlyRate: double

1. Discuss the different types of constructor that this class can have and how will you use “Super” keyword?

Ans:

1. Default constructor.
2. Constructor with essential attributes.
3. Constructor with all attributes including the additional one.
4. Write getter and setter for additional attributes and override the toString() method of Employee class which will display the all details of that particular CasualEmployee.



# Create a test class and test all classes by creating their objects and by calling their toString() method.

# 

# Create a rough class diagram for above scenario.

Ans:

|  |
| --- |
| Employee |
| - name: String  - employeeId: int  - dateOfBirth:  LocalDate  - address: String  - phoneNumber: String  - email: String  - department: String  - designation: String  - dateOfJoining: LocalDate  - salary: double  - paymentMethod: PaymentMethod  - bankDetails: BankDetails  - leaveBalance: int  - employeeType: |EmployeeType  - taxRate: double |
| |  |  | | --- | --- | | + Employee() |  |  |  |  | | --- | --- | | + Employee(String, int, String, String, double) |  |  |  |  | | --- | --- | | + Employee(String, int, LocalDate, String, String, String, String, String, LocalDate, double, PaymentMethod, BankDetails, int, EmployeeType, double) |  |  |  |  | | --- | --- | | + String getName() |  |  |  |  | | --- | --- | | + void setName(String) |  |  |  |  | | --- | --- | | + int getEmployeeId() |  |  |  |  | | --- | --- | | + void setEmployeeId(int) |  |  |  |  | | --- | --- | | + LocalDate getDateOfBirth() |  |  |  |  | | --- | --- | | + void setDateOfBirth(LocalDate) |  |  |  |  | | --- | --- | | + String getAddress() |  |  |  |  | | --- | --- | | + void setAddress(String) |  |  |  |  | | --- | --- | | + String getPhoneNumber() |  |  |  |  | | --- | --- | | + void setPhoneNumber(String) |  |  |  |  | | --- | --- | | + String getEmail() |  |  |  |  | | --- | --- | | + void setEmail(String) |  |  |  |  | | --- | --- | | + String getDepartment() |  |  |  |  | | --- | --- | | + void setDepartment(String) |  |  |  |  | | --- | --- | | + String getDesignation() |  |  |  |  | | --- | --- | | + void setDesignation(String) |  |  |  |  | | --- | --- | | + LocalDate getDateOfJoining() |  |  |  |  | | --- | --- | | + void setDateOfJoining(LocalDate) |  |  |  |  | | --- | --- | | + double getSalary() |  |  |  |  | | --- | --- | | + void setSalary(double) |  |  |  |  | | --- | --- | | + PaymentMethod getPaymentMethod() |  |  |  |  | | --- | --- | | + void setPaymentMethod(PaymentMethod) |  |  |  |  | | --- | --- | | + BankDetails getBankDetails() |  |  |  |  | | --- | --- | | + void setBankDetails(BankDetails) |  |  |  |  | | --- | --- | | + int getLeaveBalance() |  |  |  |  | | --- | --- | | + void setLeaveBalance(int) |  |  |  |  | | --- | --- | | + EmployeeType getEmployeeType() |  |  |  |  | | --- | --- | | + void setEmployeeType(EmployeeType) |  |  |  |  | | --- | --- | | + double getTaxRate() |  |  |  |  | | --- | --- | | + void setTaxRate(double) |  |  |  |  | | --- | --- | | + double calculateSalary() |  |  |  |  | | --- | --- | | + void applyLeave(int) |  |  |  |  | | --- | --- | | + void updatePersonalInfo(String, String, String) |  |  |  |  | | --- | --- | | + void changeDepartment(String) |  |  |  |  | | --- | --- | | + void promoteEmployee(String, double) |  |  |  |  | | --- | --- | | + void terminateEmployee(LocalDate) |  |  |  |  | | --- | --- | | + void generatePayslip() |  |  |  |  | | --- | --- | | + double calculateTaxDeduction() |  |  |  |  | | --- | --- | | + void updateBankDetails(BankDetails) |  |  |  |  | | --- | --- | | + void requestLeave(int) |  |  |  |  | | --- | --- | | + void printEmployeeDetails() |  |  |  | | --- | | + double calculateNetPay() | |

|  |
| --- |
| FulltimeEmployee |
| - double bonus |
| |  |  | | --- | --- | | + FulltimeEmployee() |  |  |  |  | | --- | --- | | + FulltimeEmployee(String, int, String, String, double, double) |  |  |  |  | | --- | --- | | + FulltimeEmployee(String, int, LocalDate, String, String, String, String, String, LocalDate, double, PaymentMethod, BankDetails, int, EmployeeType, double, double) |  |  |  |  | | --- | --- | | + double getBonus() |  |  |  |  | | --- | --- | | + void setBonus(double) |  |  |  | | --- | | + void printEmployeeDetails() | |

|  |
| --- |
| CasualEmployee |
| - double hourlyRate |
| |  |  | | --- | --- | | + CasualEmployee() |  |  |  |  | | --- | --- | | + CasualEmployee(String, int, String, String, double, double) |  |  |  |  | | --- | --- | | + CasualEmployee(String, int, LocalDate, String, String, String, String, String, LocalDate, double, PaymentMethod, BankDetails, int, EmployeeType, double, double) |  |  |  |  | | --- | --- | | + double getHourlyRate() |  |  |  |  | | --- | --- | | + void setHourlyRate(double) |  |  |  | | --- | | + void printEmployeeDetails() | |

|  |
| --- |
| PaymentMethod |
| Constants |
| |  |  | | --- | --- | | - BANK\_TRANSFER |  |  |  |  | | --- | --- | | - CASH |  |  |  |  | | --- | --- | | - CHEQUE |  |  |  | | --- | | - MOBILE\_PAYMENT | |

|  |
| --- |
| BankDetails |
| |  |  | | --- | --- | | - String bankName |  |  |  |  | | --- | --- | | - String accountNumber |  |  |  | | --- | | - String branch | |
| |  |  | | --- | --- | | + BankDetails(String, String, String) |  |  |  |  | | --- | --- | | + String getBankName() |  |  |  |  | | --- | --- | | + void setBankName(String) |  |  |  |  | | --- | --- | | + String getAccountNumber() |  |  |  |  | | --- | --- | | + void setAccountNumber(String) |  |  |  |  | | --- | --- | | + String getBranch() |  |  |  |  | | --- | --- | | + void setBranch(String) |  |  |  | | --- | | + String toString() | |

|  |
| --- |
| EmployeeType |
| Constants |
| |  |  | | --- | --- | | - FULLTIME |  |  |  |  | | --- | --- | | - PARTTIME |  |  |  | | --- | | - CONTRACT | |

|  |
| --- |
| TestPayrollSystem |
| + main(String[] args) |