title programmer input daily hours

loop programmer has more to enter

client->CBDcompany: getEmployee(Employee ID number)

CBDcompany-->client: programmer

client->programmer: inputDailyHour(invoice number, dailyHour)

programmer->programmer: updateWeeklyTotalHour(dailyHour)

programmer->CBDcompany: getProject(invoice number)

CBDcompany-->programmer: project

programmer->project: getStatus()

project-->programmer: status

alt total weekly hour is less than 60 and project is insolvent:

programmer->programmer: getTimeLogEntry(invoice number)

programmer->timeLogEntry: updateTimeLog(dailyHour)

timeLogEntry->timeLogEntry: addWeeklyHour(dailyHour)

end alt

timeLogEntry-->client: void

title clerical worker inputs daily hours

client->CBDcompany: getEmployee(employee ID number)

CBDcompany-->client: clerical worker

client->clerical worker: inputDailyHours(dailyHour)

clerical worker->clerical worker:addWeeklyHour(dailyHour)

clerical worker-->client: void

title end of week

client->CBDcompany: endOfWeek():

loop iterate over the employee list

CBDcompany->CBDcompany: getNextEmployee()

alt the employee is a programmer

CBDcompany->programmer: endOfWeek()

loop iterate over the time log entries list

programmer->programmer: getNextTimeLogEntry()

programmer->timeLogEntry: endOfWeek()

timeLogEntry->timeLogEntry: getWeeklyHour()

timeLogEntry->programmer: getHourlyRate()

timeLogEntry->timeLogEntry: calculateCharge(weekly hour, hourly rate)

timeLogEntry->timeLogEntry: getProject()

timeLogEntry->timeLogEntry: setWeeklyHour(0)

timeLogEntry->project: billCharge(weeklyCharge)

project->project: updateBalance(weeklyCharge)

project->programmer: void

programmer->programmer:getWeeklyHour()

programmer->programmer:getHourlyRate()

programmer->programmer: setTotalWeeklyHour(0)

programmer->CBDcompany: getpaid(weekly hour, hourly rate)

CBDcompany-->programmer: void

end loop

end alt

alt the employee is a clerical worker

CBDcompany->clerical worker: endOfWeek()

clerical worker->clerical worker: getWeeklyHour()

clerical worker->clerical worker: getHourlyRate()

clerical worker->clerical worker: getOverheadProject()

clerical worker->overheadProject: charge(weekly hour, hourly rate)

overheadProject->overheadProject: adjustBalance(weekly charge)

overheadProject-->clerical worker: void

clerical worker->clerical worker: setWeelyHour(0)

clerical worker-->CBDcompany: getPaid(weekly hour, hourly rate)

CBDcompany-->clerical worker: void

end alt

alt the employee is a salesperson

CBDcompany->salesperson: endOfWeek()

salesperson->salesperson: calculateCommission()

salesperson->salesperson: resetCommission()

salesperson->CBDcompany: getPaid(Commission)

CBDcompany-->salesperson: void

end alt

end loop

loop iterate over project list

CBDcompany->CBDcompany: getNextProject()

CBDcompany->project: endOfWeek()

project->project: getStatus()

alt status is stopped or ended

project->project: setStatus(closed)

project->project: getBalance()

project->project: getCustomer()

project->customer: returnFund(balance)

end alt

alt balance is not positive

project->project: setStatus(insolvent)

project->customer: alert()

customer-->project: void

end alt

project-->CBDcompany: void

end loop

title Print All Programmer Weekly Time Log Report

loop iterate over the programmer list

client->CBDcompany: getNextProgrammer()

CBDcompany-->client: programmer

client->programmer: weeklyReport:

programmer->programmer: getName()

programmer-> programmer: getHourlyRate()

programmer->programmer: printInfo(name, rate)

loop iterate over the time log

programmer->programmer:getNextTimeLogEntry()

programmer->TimeLog:printDate()

TimeLog-->programmer:void

programmer->TimeLog:getCustomerProject()

TimeLog-->programmer: customerProject

programmer->customerProject:printCstName()

customerProject-->programmer: void

programmer->customerProject:printInvoiceNum()

customerProject-->programmer: void

programmer->customerProject:printCunstomer()

customerProject->customer: printCstName()

customer-->customerProject: void

customerProject-->programmer: void

programmer->TimeLog:printTotalWeeklyHour()

TimeLog-->programmer:void

programmer-->client:void

end loop

end loop

title weekly Employee Earning Reports

client->CBDcompany: getEmployee(Employee ID number)

CBDcompany-->client: programmer

CBDcompany->programmer:getProject()

programmer-->CBDcompany:Project

CBDcompany->Project:getTotalWeeklyCharge()

Project->timeLogEntry:getTotalWeeklyHours()

timeLogEntry-->Project:TotalWeeklyHours

Project->timeLogEntry:getDate ()

timeLogEntry-->Project:Date

Project->programmer: getHourlyRate()

programmer-->Project:Hourly Rate

Project->Project:calculateTotalWeeklyCharge()

Project-->CBDcompany:TotalWeeklyCharge

CBDcompany-> Project:getCustomer()

Project-->CBDcompany:Customer

CBDcompany-> CBDcompany:PrintTheEarning()

CBDcompany-->client: void

Class project