Basics of Software Engineering

Lab 7

245693

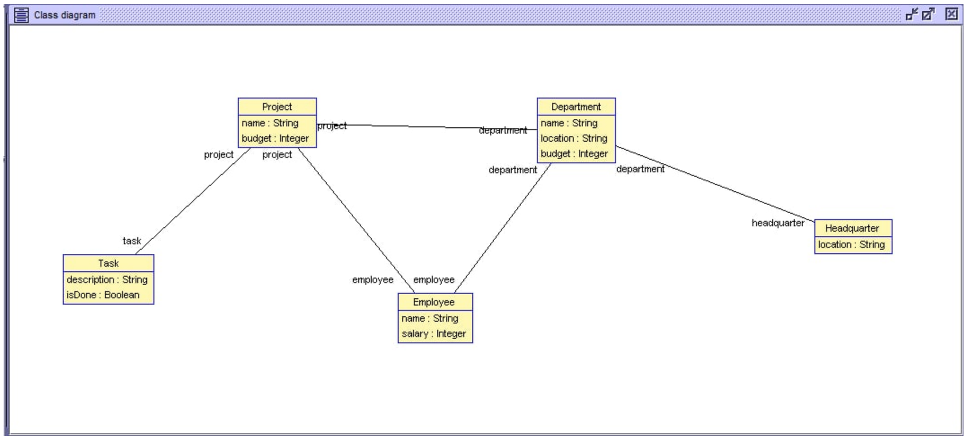
Task 1

```
-- $ProjectHeader: use 0.393 Wed, 16 May 2007 14:10:28 +0200 opti $

model Company

-- classes

class Employee
attributes
  name : String
  salary : Integer
end

class Department
attributes
  name : String
  location : String
  budget : Integer
end

class Project
attributes
  name : String
  budget : Integer
end
```

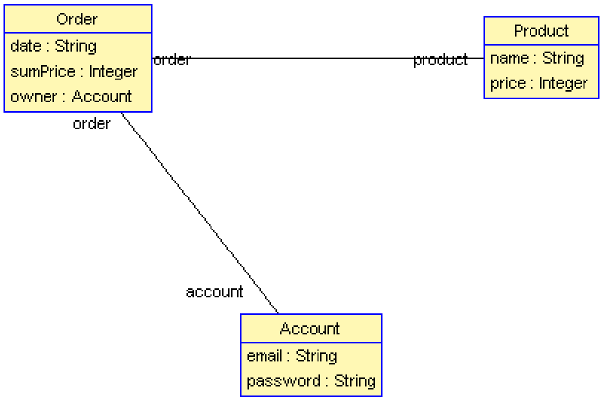
```
class Task
attributes
description : String
isDone : Boolean
end

class Headquarter
attributes
location : String
end


-- associations

association WorksIn between
  Employee[*]
  Department[1..*]
end

association WorksOn between
  Employee[*]
  Project[*]
end

association Controls between
  Department[1]
  Project[*]
end
```

```
association Located between
  Department[1]
  Headquarter[1]
end

association Has between
Project[1]
Task[*]
end

-- OCL constraints

constraints

context Department
    -- the number of employees working in a department must
    -- be greater or equal to the number of projects
    -- controlled by the department
  inv MoreEmployeesThanProjects:
    self.employee->size >= self.project->size

context Headquarter
-- location size
inv HeadquarterLocation:
self.location.size() >= 1

context Employee
    -- employees get a higher salary when they work on
    -- more projects
```

```
 inv MoreProjectsHigherSalary:
  Employee.allInstances->forAll(e1, e2 |
   e1.project->size > e2.project->size
    implies e1.salary > e2.salary)



context Project
  -- the budget of a project must not exceed the
  -- budget of the controlling department
 inv BudgetWithinDepartmentBudget:
  self.budget <= self.department.budget

  -- employees working on a project must also work in the
  -- controlling department
 inv EmployeesInControllingDepartment:
  self.department.employee->includesAll(self.employee)

context Task
inv DescriptionLength:
self.description.size() >= 1
```



Task 2



```
-- $ProjectHeader: use 0.393 Wed, 16 May 2007 14:10:28 +0200 opti $

model Software

-- classes

class Account
attributes
  email : String
  password : String
end

class Order
attributes
date : String
sumPrice : Integer
owner : Account
end

class Product
attributes
name : String
price : Integer
end
-- associations

association AssignedTo between
  Account[1]
  Order[*]
end

association Assigned between
  Order[1]
  Product[*]
end

-- OCL constraints
```

constraints

```
context Account

--password contain non empty password which contains at least 1 digit and 1 special character ($ # @)

inv PasswordValidation:

self.password.size() > 1



context Product

--Product price must be greater than zero

inv PriceValidation:

self.price > 0
```