

Accountant

Method Header	Description
public Accountant(String name)	Should start with a bonus budget of 0 and no team they are officially supporting
<pre>super (name) ; this.setBonusBudget (0); //Set the budget amount as zero</pre>	
public TechnicalLead getTeamSupported()	Should return a reference to the TechnicalLead that this Accountant is currently supporting. If they have not been assigned a TechnicalLead null should be returned
<pre>// Return the team supported. // Support team will be assigned by the business lead // while addReport return teamSupported;</pre>	
public void supportTeam(TechnicalLead lead)	Should allow a reference to a TechnicalLead to be passed and saved. Once this happens the Accountant's bonus budget should be updated to be the total of each SoftwareEngineer's base salary that reports to that TechnicalLead plus 10%. For example, if the TechnicalLead supports 2 SoftwareEngineers, each with

Method Header	Description
	a salary of 75000, the Accountant's budget should be 150000 + 15000 for a total of 165000
<pre> this.teamSupported = lead; // set the bonus budget as sum of technical lead's team salary plus 10 % double budget = 0; for (SoftwareEngineer e: lead.getTeam()){ //iterate the TechnicalLead's team budget += e.getBaseSalary(); // add their base salary } budget += budget* 1.1; // 10% = .1, 100% = 1 this.setBonusBudget (budget); </pre>	
public boolean approveBonus(double bonus)	Should take in a suggested bonus amount and check if there is still enough room in the budget. If the bonus is greater than the remaining budget, false should be returned, otherwise true. If the accountant is not supporting any team false should be returned.
<pre> //Check the requested bonus amount is within budget? //if yes then return true else return false if (bonus <= this.getBonusBudget()) return true; else return false; </pre>	

Method Header	Description
public String employeeStatus()	Should return a String representation of this Accountant that includes their ID, name, the size of their currently managed budget and the name of the TechnicalLead they are currently supporting. Example: "1 Kasey with a budget of 22500.0 is supporting Satya Nadella"
<pre>String <u>result</u>; <u>result</u> = super.employeeStatus(); <u>result</u> += " supporting " + teamSupported.getName(); return result;</pre>	

BusinessLead

Method Header	Description
public BusinessLead(String name)	Should create a new BusinessLead that is a Manager. The BusinessLead's base salary should be twice that of an Accountant. They should start with a head count of 10.

Method Header	Description
<pre>// The BusinessLead's base salary should be twice of an Accountant. // Set initial headCount as 10. super(name); this.setBaseSalary(getBaseSalary()*2); this.headCount=10;</pre>	
public boolean hasHeadCount()	Should return true if the number of direct reports this manager has is less than their headcount.
public boolean addReport(Accountant e, TechnicalLead supportTeam)	Should accept the reference to an Accountant object, and if the BusinessLead has head count left should add this employee to their list of direct reports. If the employee is successfully added to the BusinessLead's direct reports true should be returned, false should be returned otherwise. Each time a report is added the BusinessLead's bonus budget should be increased by 1.1 times that new employee's base salary. That employee's team they are supporting should be updated to reflect the reference to the TechnicalLead given. If the employee is successfully added true should be returned, false otherwise.

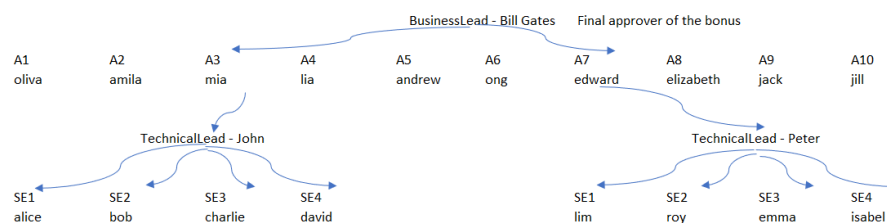
Method Header	Description
<pre> // if BusinessLead has headCount // add accountant to BusinessLead's team // set the BusinessLead as Accountant's manager // set the TechnicalLead as Accountant's support team // set Accountant as TechnicalLead to request bonus // increase the budget amount by 1.1 times // of the new employee's base salary // return true // else // return false if (hasHeadCount()) { team.add(e); // Add accountant to team e.setManager(this); // set Accountant's manager e.supportTeam(supportTeam); // Set Accountant's teamSupported supportTeam.setAccountant(e); // Set the accountant of the technical lead this.setBonusBudget(this.getBaseSalary() + (e.getBaseSalary()*1.1)); // Increase the budget return true; } else { return false; } </pre>	
<p>public boolean requestBonus(Employee e, double bonus)</p>	<p>Should check if the bonus amount requested would fit in current BusinessLead's budget. If it is, that employee should get that bonus, the BusinessLeader's budget should be deducted and true should be returned. False should be returned otherwise</p>

Method Header	Description
<pre> // if requested bonus amount within BusinessLead's budget // deduct the bonus amount from budget // return true // else // return false </pre> <pre> if (bonus <= this.getBonusBudget()) { this.setBonusBudget(this.getBonusBudget() - bonus); return true; } else { return false; } </pre>	
<pre> public boolean approveBonus(Employee e, double bonus) </pre>	<p>This function should look through the Accountants the BusinessLead manages, and if any of them are supporting a the TechnicalLead that is the manager of the Employee passed in then the Accountant's budget should be consulted to see if the bonus could be afforded. If the team can afford the bonus it should be rewarded and true returned, false otherwise</p>

Method Header	Description
<pre> // if employee's TechnicalLead is supported by the Accountant // And the Accountant is reporting to this BusinessLead // check the Accountant can approveBonus // deduct the bonus amount from Accountant's budget // return true // else false TechnicalLead tl = (TechnicalLead) e.getManager(); Accountant accountant = tl.getAccountant(); if (team.contains(accountant)){ if (accountant.approveBonus(bonus)) { accountant.setBonusBudget(accountant.getBonusBudget() - bonus); return true; } } return false; </pre>	

Accountants are responsible to manage the teams budget

Technical Leads lead the team of software engineers to develop the project



Try to test the implementation by setting the code access, request bonus etc.,