*Софийски университет „Св. Климент Охридски“*    
 *Факултет по математика и информатика*

*Курс по Структури от данни и програмиране специалност Информатика*

**Управление на фирма**

*Автор: Марина Стоименовска*

**Съдръжание**

**Глава 1. Увод**

* 1. **Описание и идея на проекта**

Управлението на фирма не е лесна работа. Именно в това се убеждава и шефът на голяма ИТ компания. Освен с много документи администрацията и шефът на една фирма за затрупани и със разпределяне на служителите в екипи.

Изглежда просто когато потокът на хора е малък, но поради бързото разрастване е назначаването и повишаването, както и уволняването на служители от фирмата шефът има нужда от помощ, всичко може да почака, но не и хората.

Идеята на проекта е да разреши проблема на шефа и фирмата му при голям поток от служители. Проекта бързо разпределя кой на кого е подчинен и кои са неговите подчинени. В решението има няколко функционалности които ще помогнат на шефа в различни ситуации, както при напускащи и новопостъпващи служители, така и при повишение на настоящи служители.

* 1. **Цел и задачи на разработката**

Основната цел е да се използва подходяща структура от данни за реализация на проект която в същото време ще е разбираема и удобна за използване от страна на шефа на ИТ компанията. За решаване на задачата ще се използва структура от данни дърво с променлив брой на върхове.

За проекта ще се реализират 2 основни класа. В зависимост от нуждите компанията, преструктурирнето може да настъпи след:

* Назначаване на нов служител
* Повишение на настоящ служител
* Уволнение на стар служител
  1. **Структура на документацията**

Структурата е разделена на 4 основни нива в които са описани етапите през които преминава проекта.

В първата час е описана идеята, целите и задачите за разработка, които трябва да бъдат реалицирани в проекта.  
Втората част е посветена на основните дефиниции, класове, методи и алгоритми в проекта.

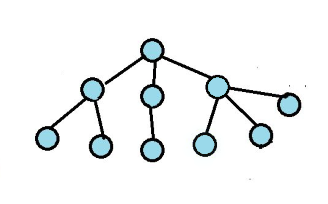
Третата част е описана архитектурата заедно с примери и от кода.

Четвъртата се грижи за реализация, оптимизация и тестване на проекта.

**Глава 2. Преглед на предметната област**

**2.1. Основни дефиниции, концепции и алгоритми, които ще бъдат използвани**

Във всяка една компания, без значение колко е малка или голяма, съществува някакъв тип йерархия. В голямата ИТ компания йерархията е голяма е проста: всеки е подчинен на някого, а всички са подчинени на шефа. Броят на подчинените е неограничен.

****

*Фигура 1. Йерархия на компанията*

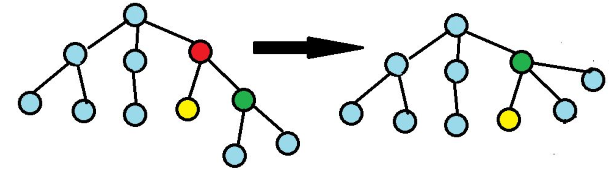
За нуждите на компанията ще се реализират 5 основни фукционалности, които ще покрият всички случаи относно преструктурирането на йерархията:

Първата функционалност е назначаване на нов служител, който не е шеф на никого – това е най-лесният случай. В него се описва кой е новият служител и съответно кой е неговият шеф.

Втората функционалност е назначаване на нов служител и прехвърляне му на подчинени – с тази функция, шефът има възможност да назначи нов служител, на който му съпоставя подчинени, като ги описва поименно и разбира се на кого е подчинен.

Третата функционалност е уволняване на служител, който не е шеф на никого – тази фунционалност може да се използва и при напускащи служители, при пенсиониране, както и при съкращаване на служителите. При нея също като в първата, премахва се служителя, без никакви допълнителни размествания, тъй като той няма подчинени.

Може би една от най-важните функционалности е уволняване на служител, който ръководи други служители – в нея служителят който си отива трябва да бъде заместен от някой от подчинените му достоен за мястото. Това е служителят който има най-много подчинени, а при равен брой на подчинени, няма значение кой ще заеме мястото.

  
  
 *Фигура 2. Уволняване на служител, който ръководи други служители*

Последната, но не и по-важност функционалност е повишаване на служител, прехвърляйки му подчинени – тук се описва кой е служителят който се повишава, кой е шефът му и кои подчинени се прехвърлят под раководството му.

Освен от манипулиране на йерархията, шефът се нуждае и от информация относно текущото състояние на структурата във фирмата. В зависимост от нуждите има 3 типа информация която може да даде:

* Кои са преките подчинени на даден служител.
* Кои са всички подчинени на даден служител, подредени по нива.
* Кой е прекият шеф на даден служител (шефът е шеф сам на себе си).

Проекта реализира преструктуриране на йерархия на ИТ компания, като шефа въвежда промените чрез клавиатурата. Самата структура може да се види на екрана след съответната команда.

За изпълнението на задачата не са взети в предвид следните фактори:  
Служителите с едно и също име не може да има.  
Компанията няма идентификатори, тъй като задачата е за една компания.  
Приемаме, че въведената информация за шеф и подчинени е коректна и тези хора работят в компанията.  
При повишение на служител, приемаме, че данните са коректни и човека е настоящ служител във фирмата и прекият му шеф не е шефът на фирмата.

**2.2. Дефиниране на проблеми и сложност на поставената задача**Задачата предоставя много на брой проблеми:  
- Шефът не може да бъде премахнат  
- Може да се назначи нов човек, който да вземе подчинените от произволни под екипи, който да има други подчинени.  
- Броят на подчинените е неограничен, заради това не може да използваме структура от данни двоично дърво.

Програмата приема команди относно преструктурирането на фирмата и във всеки един момент може да даде информация за нея.

**2.3. Подходи, методи за решаване на проблемите**За решаване на проблемите с паметта в проекта използваме инструменти на обектно-ориентираното програмиране. За представянето на данните в проекта използваме структура от данни дърво. Енкапсулацията от своя страна осигурява защита от потребителите, които не са наясно с използването на софтуера.

**2.4 Потребителски изисквания и качествени изисквания**

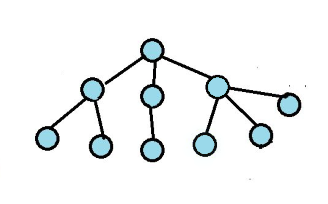
Всички данни входни/изходни се осъществяват с помощта на команди от потребителя.  
Входните данни трябва да са във вид:  
 1. Команда ако искаме да променим йерархията

2. Команда ако искаме да видим йерахрията

**Глава 3. Проектиране**

**3.1. Обща архитектура – СДП/ООП дизайн**

*Фигура 1. Йерархия на класовете*

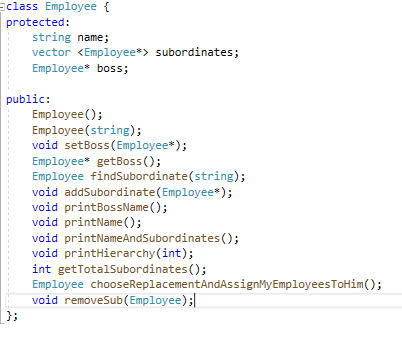
****Company и Employee са отделни класове. Класът Company има указател към променлива от тип Employee, която се грижи за шефа на фирмата. Чрез този указател, може да стигнем до всеки един служител на компанията.

*Фигура 2. Структура от данни*

Използваме структура от данни дърво с неограничен брой върхове. Върховете(подчинените) са преставени чрез вектор.

**Глава 4. Реализация,**

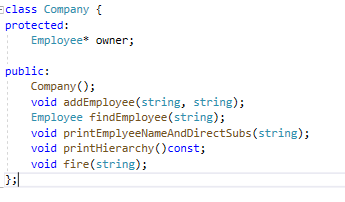
**4.1. Реализация на класове**



Клас Employee има идентификатори име, подчинени и шеф на когото е подчинен служителя на фирмата.

*Клас 1. Employee*

В клас Company е реализирана основната идея на  
Проверката за коректно въведени данни от потребителя се извършва в main-a.



*Клас 2. Company*

**4.3. Планиране, описание и създаване на тестови сценарии**

**Глава 5. Git хранилище**

**5.1. Референция към Git хранилище**

* <https://github.com/stoimenovska/sdp1920>

**Използвана литература**

[1]<https://www.geeksforgeeks.org/generic-tree-level-order-traversal/>

[2]