



Marmara University
Faculty of Engineering

CSE460

Principles of Programming Languages
Project Report#1

COURSE LECTURER:

Assistant Professor Fatma Corut Ergin

Prepared by

Name: Ahmet

Surname: Özlü

#ID: 150711047

This project developed by using *Python 3.0 on PyCharm IDE* .

This is the Person.py,

```
#!/usr/bin/python

class Person(object):

    'Common base class for all employees'
    person = 0

    def __init__(self, name=None, surName=None, gender=None, birthDate=None, deadDate=None, father=None, mother=None, children=0):
        self.name = name
        self.surName = surName
        self.gender = gender
        self.birthDate = birthDate
        self.deadDate = deadDate
        self.father = father
        self.mother = mother
        self.children = children
        Person.person += 1

    def displayCount(self):
        print("Total Person %d" % Person.person)

    def displayPerson(self):
        print("Name : ", self.name, ", Surname: ", self.surName, ", Father: ", self.father, ", Mother: ", self.mother, ", Children: ", self.children)
```

This is console screenshot,

```
1-)Ask relation
2-)Add/Delete/Update person
3-)Get information of any person
4-)Print the family tree
5-)Terminate the program
Please choose an operation!
```

By entering your choice, you can interact with the program.

You can ask relation, add/delete/update person or more.

For example, you can print the family tree by entering 4. This is screenshot for printing family tree,

```
***You chose 'Print the family tree'***
***LEVEL-3***
Sıdıka Özlü & Hasan Özlü
***LEVEL-2***
Gülşen Özlü & Aziz Özlü
Nafiye Şişli & Nazif Şişli
Neriman Karabag & Ekrem Karabag
***LEVEL 1***
Öznur Tanış & Hakan Tanış
İlknur Karabag & Abdullah Karabag
Filiz Karabag & Mehmet Karabag
Merve Özlü & Ahmet Özlü
Hatice Özlü & İlker Özlü
***LEVEL-0***
Elif Tanış
Yiğit Tanış
Nurullah Karabag
Büşra Karabag
Numan Karabag
Fatma Keser
Selami Keser
Kuzey Özlü
Melih Özlü
```

You can get information about any person,

```
***You chose 'Get information of any person'***
Enter the person name: Aziz
Enter the person surname: Özlü
Name : Aziz , Surname: Özlü , Father: Hasan , Mother: Sıdıka , Children: 4
Age: 72
ALIVE or DEAD: DEAD
Total child count: 4
Level in the tree: 2
```

And this is the printTree function,

```

def printTreeAlternatively(self):
    for tempPerson in self.peopleList:
        a = self.getLevel(self, tempPerson)
        if (a == 0):
            self.level0.append(tempPerson)
        elif (a == 1 and tempPerson.gender == "f"):
            self.level1.append(tempPerson)
        elif (a == 2 and tempPerson.gender == "f"):
            self.level2.append(tempPerson)
        elif (a == 3 and tempPerson.gender == "f"):
            self.level3.append(tempPerson)
        elif (a == 4 and tempPerson.gender == "f"):
            self.level4.append(tempPerson)

    print("****LEVEL-3 / GRANDMOTHER & GRANDFATHER****")
    for tempPerson in self.level3:
        relation = self.getChild(self, tempPerson, self.list1)

        if (len(Tree.list1) >= 1):
            if (tempPerson.gender == "f"):
                child = self.getPerson(self, self.list1[0].name, self.list1[0].surName)
                del Tree.list1[:]
                print(
                    tempPerson.name + " " + tempPerson.surName + " & " + child.father + " " + tempPerson.surName)

    print("****LEVEL-2 / CHILDREN****")
    for tempPerson in self.level2:
        relation = self.getChild(self, tempPerson, self.list1)

        if (len(Tree.list1) >= 1):
            if (tempPerson.gender == "f"):
                child = self.getPerson(self, self.list1[0].name, self.list1[0].surName)
                del Tree.list1[:]
                print(tempPerson.name + " " + tempPerson.surName + " & " + child.father + " " + tempPerson.surName)

    print("****LEVEL 1 / GRANDCHILD****")
    for tempPerson in self.level1:
        relation = self.getChild(self, tempPerson, self.list1)

        if (len(Tree.list1) >= 1):
            if (tempPerson.gender == "f"):
                child = self.getPerson(self, self.list1[0].name, self.list1[0].surName)
                del Tree.list1[:]
                print(tempPerson.name + " " + tempPerson.surName + " & " + child.father + " " + tempPerson.surName)

    print("****LEVEL-0****")
    for tempPerson in self.level0:
        if (tempPerson.name and tempPerson.surName):
            print(tempPerson.name + " " + tempPerson.surName)
        elif (tempPerson.name):
            print(tempPerson.name)
        elif (tempPerson.surName):
            print(tempPerson.surName)

```

The source code of my project attached the mail, the program can be examined by using these source code and project files.