HACETTEPE UNIVERSITY

FACULTY OF ENGINEERING

DEPARTMENT OF ARTIFICIAL INTELLIGENCE ENGINEERING

BBM203 – DATA STRUCTURE

PROGRAMMING ASSIGNMENT II

Mehmet Berat Ersarı

21992943

Due Date: 30.11.2021 (23.59)

Programming Language: C++

Subject: LinkedLists

1- INTRODUCTION

A public institution needs a “Recording System” to record the employees’ information. Because recording in the ledger is very useless. I developed a program to solve this problem. You can add an employee, delete an employee, list all employees or list all employees after a spesific appointment date etc. Also my program is too quick while making these operations.

2- METHOD

I split project into a lot of functions to make operations easilly. Also I found a few ways, thanks to these ways most functions similiar each others. I expalined in more details in the design part.

3- DEVELOPMENT

3.1- PLAN

Firstly, I defined the Temporary Class, Permanent Class and Date Class. Then, the required functions(getter&setter and << operator for Employee Class, overloading operators for Date Class (<,>,==)).Secondly, I defined CircularArrayLinkedList Class and DoubleDynamicLinkedList Class and function which associated with these classes. These functions are print functions, sort functions, insert functions, delete functions etc. for each classes. Then, I created functions for each commands and used it in Main class.

3.2- ANALYSIS

Before the started to develop project. I think which functions should I use to solve this problem easily and I did some analysis and wrote them on paper. After these, I developed my project on IDE.

3.2- DESIGN

I thought most possible edge cases and developed my code according to this to avoid overflow. I chosen some sorting algorthims such as bubble sort, selection sort. And i have print function for each command. Printing is so quick. But My code lost time while sorting. Bubble sort has O(n^2) time complexity. Maybe any other sorting alghortims can be choosen to sort. Maybe quick sort or merge sort. But I could not implement these to my code. I have NodeTemporary, DoubleDynamicLinkedList, CircularArrayLinkedList, Employee, TemporaryEmployee, Permanent Employee classes and header files. TemporaryEmployee and PermanentEmployee Classes are subclasses of Employee class. CircularArrayLinkedList class is a circular linked list with array implementation. This CircularLinkedList has Temporary Employees. Maximum size of this is twenty. This has insert, delete, sort methods. DoubleDynamicLinkedList has a dynamic structure. This takes Permanent Employess. There is no limit for size and this has insert, delete and sort methods.

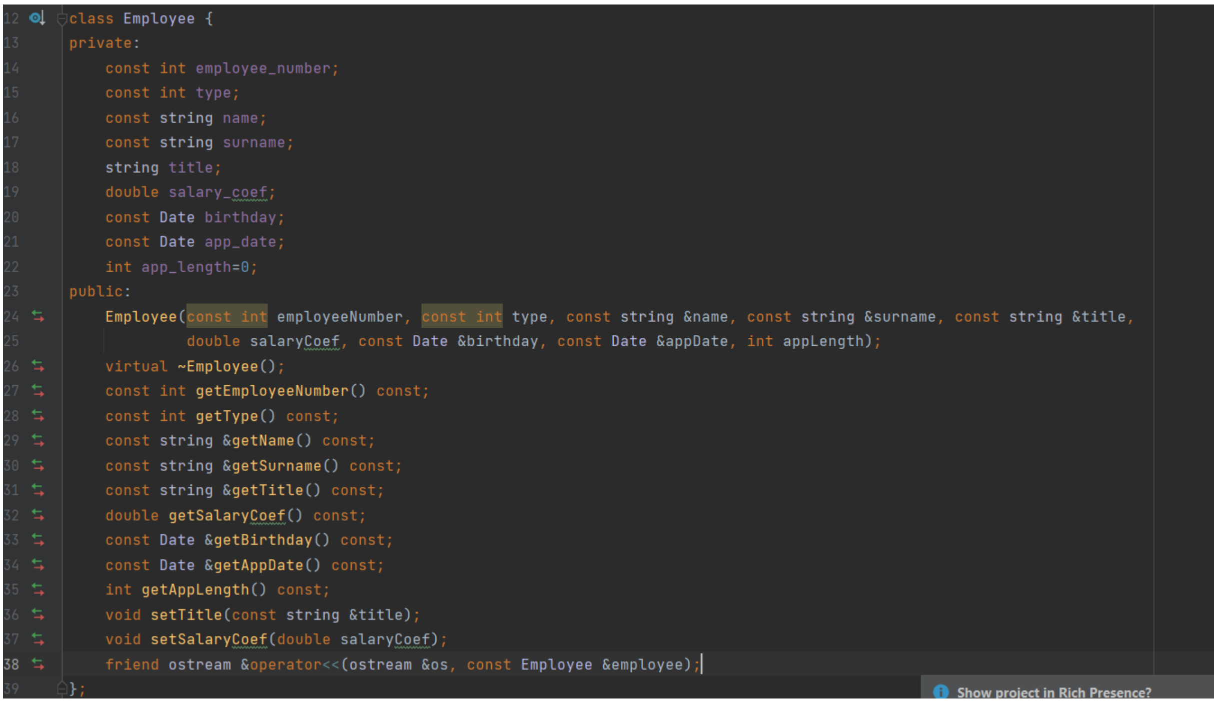
3.3- IMPLEMENTATION

I can explain the commands’ methods. For example command 6, this want me to sort in ascending order by employee number. CircularArrayLinkedList already in sorted order. I sort the DoubleDynamicLinkedList and go in comman6. In this method, there is a for loop and it executes sum of size of the lists times. And print the employees while by comparing elements of two list. Other command methods has similiar logic with this method.

3.4- PROGRAMMER CATALOG

I spent about 30-40 hours on this Project. I spent 3-4 hours for analysis, 20-25 hours for design and implementation, 2-3 hours for testing, 5-6 hours for reporting. Other programmers can use my code whenever they want. I will upload my code on github. My insert function for CircularArrayLinkedList is a bit complex structure, they may rewrite this part again. If they want to sort by different things. They can add new sorting alghoritms. In main class there are command functions from 6 to 12. I used it while taking input. Now, I will explain all header files and theirs functions.

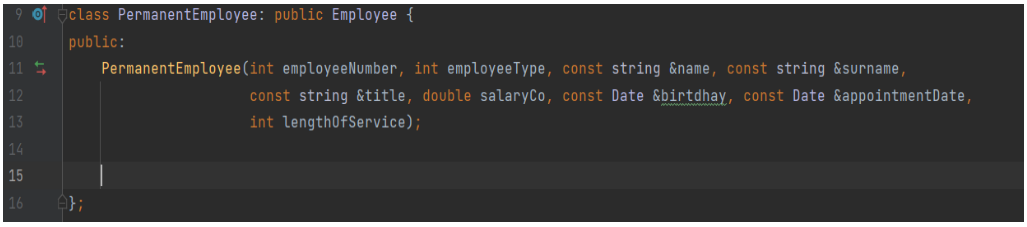
Employee.h



```
class Employee {
private:
    const int employee_number;
    const int type;
    const string name;
    const string surname;
    string title;
    double salary_coef;
    const Date birthday;
    const Date app_date;
    int app_length=0;
public:
    Employee(const int employeeNumber, const int type, const string &name, const string &surname, const string &title,
             double salaryCoef, const Date &birthday, const Date &appDate, int appLength);
    virtual ~Employee();
    const int getEmployeeNumber() const;
    const int getType() const;
    const string &getName() const;
    const string &getSurname() const;
    const string &getTitle() const;
    double getSalaryCoef() const;
    const Date &getBirthday() const;
    const Date &getAppDate() const;
    int getAppLength() const;
    void setTitle(const string &title);
    void setSalaryCoef(double salaryCoef);
    friend ostream &operator<<(ostream &os, const Employee &employee);
};
```

Employee.h has employee number, employee type (Temporary employee as 0, Permanent employee as 1), name, surname, title, salary_coef, birthday, app_date and app_length. Title, salary_coef and app_length can be change in this project. That is why they aren't constant. Also, it has getter and setter methods. Also has overload operator "<<". This is used to print all information of an employee.

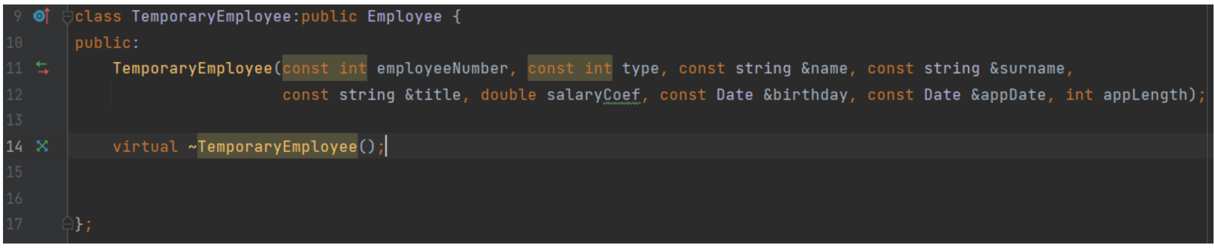
PermanentEmployee.h



```
class PermanentEmployee: public Employee {
public:
    PermanentEmployee(int employeeNumber, int employeeType, const string &name, const string &surname,
                      const string &title, double salaryCo, const Date &birtdhay, const Date &appointmentDate,
                      int lengthOfService);

};
```

PermanentEmployee.h is a subclass of Employee.h. It has all methods and members Employee.h has.

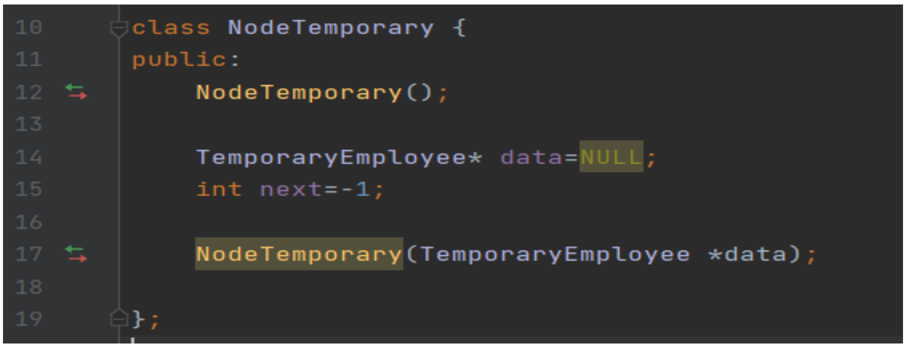
TemporaryEmployee.h



```
class TemporaryEmployee:public Employee {
public:
    TemporaryEmployee(const int employeeNumber, const int type, const string &name, const string &surname,
                      const string &title, double salaryCoef, const Date &birthday, const Date &appDate, int appLength);

    virtual ~TemporaryEmployee();


};
```

TemporaryEmployee.h is a subclass of Employee.h. It has all methods and members Employee.h has.

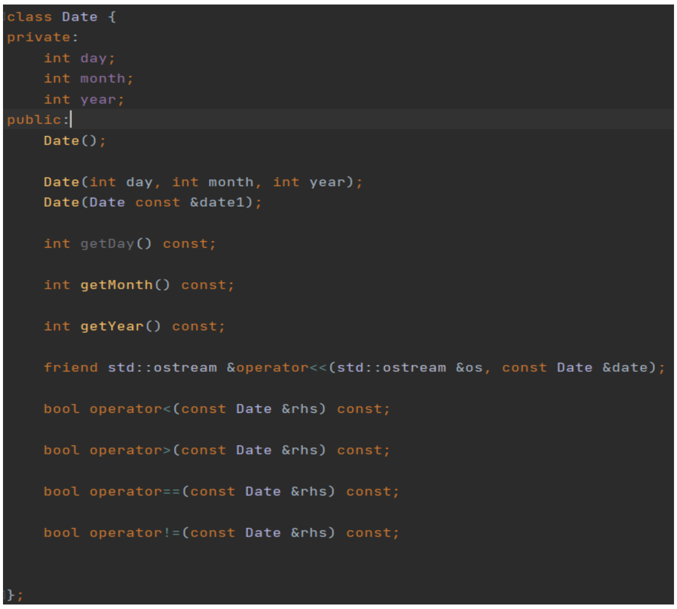
NodeTemporary.h



```
class NodeTemporary {
public:
    NodeTemporary();

    TemporaryEmployee* data=NULL;
    int next=-1;

    NodeTemporary(TemporaryEmployee *data);

};
```

NodeTemporary.h is used to store TemporaryEmployee object and index of next value in CircularArrayLinkedList.

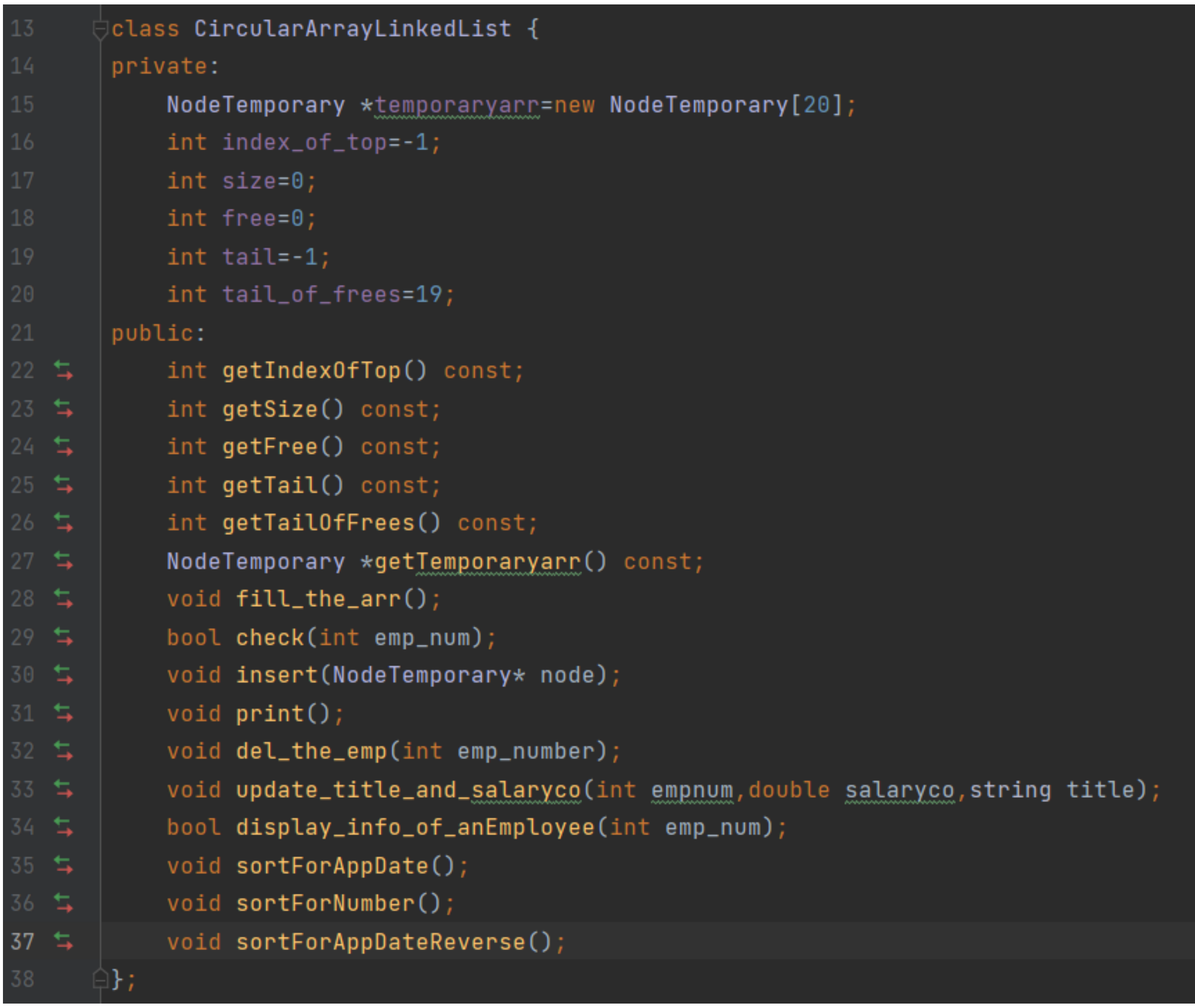
Date.h



```
class Date {
private:
    int day;
    int month;
    int year;
public:
    Date();

    Date(int day, int month, int year);
    Date(Date const &date1);

    int getDay() const;

    int getMonth() const;

    int getYear() const;

    friend std::ostream &operator<<(std::ostream &os, const Date &date);

    bool operator<(const Date &rhs) const;

    bool operator>(const Date &rhs) const;

    bool operator==(const Date &rhs) const;

    bool operator!=(const Date &rhs) const;


};
```

Date.h is used to store date of birthday and appointment date. It has overloading operator to compare two dates and getter methods.

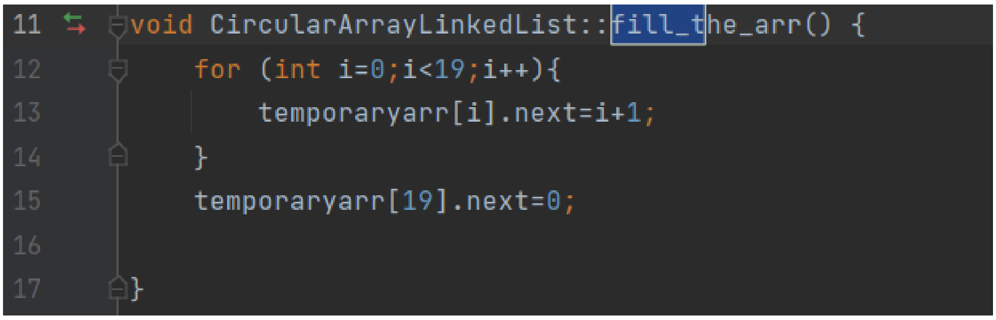
CircularArrayLinkedList.h



```
class CircularArrayLinkedList {
private:
    NodeTemporary *temporaryarr=new NodeTemporary[20];
    int index_of_top=-1;
    int size=0;
    int free=0;
    int tail=-1;
    int tail_of_frees=19;
public:
    int getIndexOfTop() const;
    int getSize() const;
    int getFree() const;
    int getTail() const;
    int getTailOfFrees() const;
    NodeTemporary *getTemporaryarr() const;
    void fill_the_arr();
    bool check(int emp_num);
    void insert(NodeTemporary* node);
    void print();
    void del_the_emp(int emp_number);
    void update_title_and_salaryco(int empnum,double salaryco,string title);
    bool display_info_of_anEmployee(int emp_num);
    void sortForAppDate();
    void sortForNumber();
    void sortForAppDateReverse();
};
```

CircularArrayLinkedList.h is used to store NodeTemporary objects. It has different members. “temporaryarr” is an array with size of 20. It stores the NodeTemporaryObjects. İndex_of_top is first object of the array. Tail is the last object of array. Next of the tail object is index_of_top. Free is the first free index of the array. Tail_of_frees is the last free index of the array. Also, this has getter methods.

void fill_the_arr();



```
void CircularArrayLinkedList::fill_the_arr() {
    for (int i=0;i<19;i++){
        temporaryarr[i].next=i+1;
    }
    temporaryarr[19].next=0;

}
```

This function initiliazes the array and connects objects.

void check(int emp_num):

This function checks whether an element exists. If it exists, it returns true. Otherwise, returns false.

void insert(NodeTemporary* node):

This function adss an element in ordered by employee_number. If size reaches maximum size(20), it prints “Error! There is no more empty field.”

void print():

This function prints all Employee with all information in the array.

void del_the_emp(int emp_num):

This function deletes the Employee with given number emp_num.

void update_title_and_salaryco(int emp_num, string title, double salary_co):

This function updates title and salary coefficient with given Employee number.

void display_info_of_anEmployee(int emp_num):

This function finds the Employee with given number. Then, prints all information.

void sortForAppDate():

This function sorts the arr ordered by AppointmentDate.

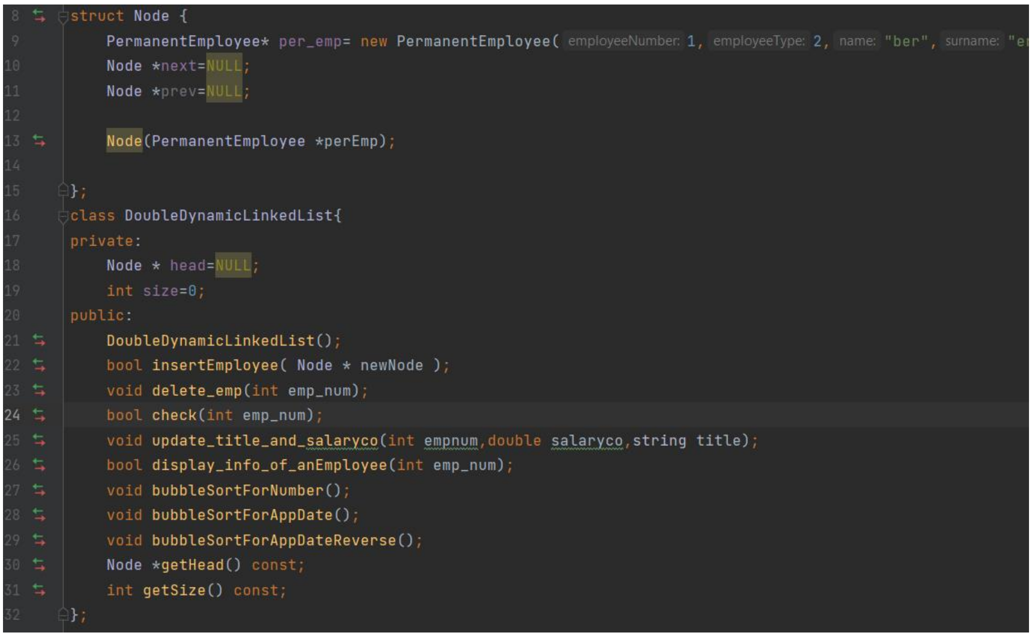
void sortForNumber():

This function sorts the arr ordered by Number.

void sortForAppDateReverse():

This function sorts the arr ordered by AppointmentDate in descending order.

DoubleDynamicLinkedList.h



```cpp
struct Node {
    PermanentEmployee* per_emp= new PermanentEmployee( employeeNumber: 1, employeeType: 2, name: "ber", surname: "e
    Node *next=NULL;
    Node *prev=NULL;

    Node(PermanentEmployee *perEmp);

};
class DoubleDynamicLinkedList{
private:
    Node * head=NULL;
    int size=0;
public:
    DoubleDynamicLinkedList();
    bool insertEmployee( Node * newNode );
    void delete_emp(int emp_num);
    bool check(int emp_num);
    void update_title_and_salaryco(int empnum,double salaryco,string title);
    bool display_info_of_anEmployee(int emp_num);
    void bubbleSortForNumber();
    void bubbleSortForAppDate();
    void bubbleSortForAppDateReverse();
    Node *getHead() const;
    int getSize() const;
};
```

DoubleDynamicLinkedList.h is used to store Node objects. It has different members. Size is the size of the LinkedList. Head is the first object of the LinkedList. Also it has a few getter methods and functions.

void insertEmployee(Node* newNode):

This function adss an element in ordered by AppointmentDate.

void delete_emp(int emp_num):

This function deletes the Employee with given number emp_num.

void check(int emp_num):

This function checks whether an element exists. If it exists, it returns true. Otherwise, returns false.

void update_title_and_salaryco(int emp_num, string title, double salary_co):

This function updates title and salary coefficient with given Employee number.

void display_info_of_anEmployee(int emp_num):

This function finds the Employee with given number. Then, prints all information.

void bubbleSortForNumber():

This function sorts the LinkedList ordered by Number.

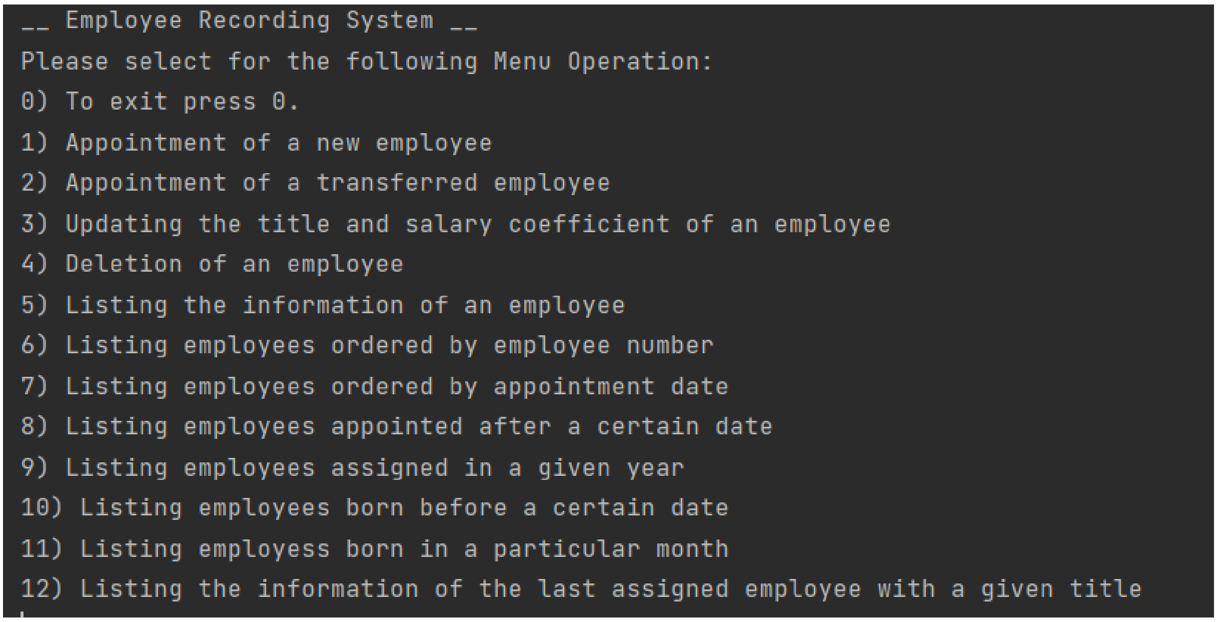
void bubbleSortForAppDate():

This function sorts the LinkedList ordered by AppointmentDate,

void bubbleSortForAppDateReverse():

This function sorts the LinkedList ordered by AppointmentDate in descending order.

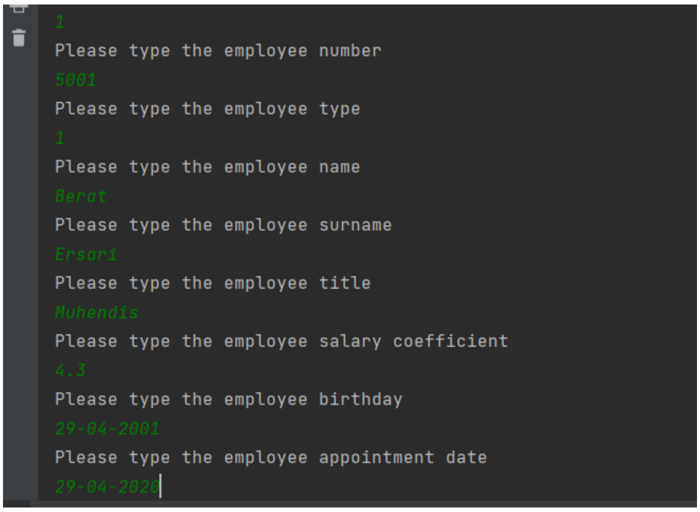
3.5- USER CATALOG



```
__ Employee Recording System __
Please select for the following Menu Operation:
0) To exit press 0.
1) Appointment of a new employee
2) Appointment of a transferred employee
3) Updating the title and salary coefficient of an employee
4) Deletion of an employee
5) Listing the information of an employee
6) Listing employees ordered by employee number
7) Listing employees ordered by appointment date
8) Listing employees appointed after a certain date
9) Listing employees assigned in a given year
10) Listing employees born before a certain date
11) Listing employess born in a particular month
12) Listing the information of the last assigned employee with a given title
```

When you open the program, you will see the screen above. If you want to exit, press 0. I explained all commands seperately below.

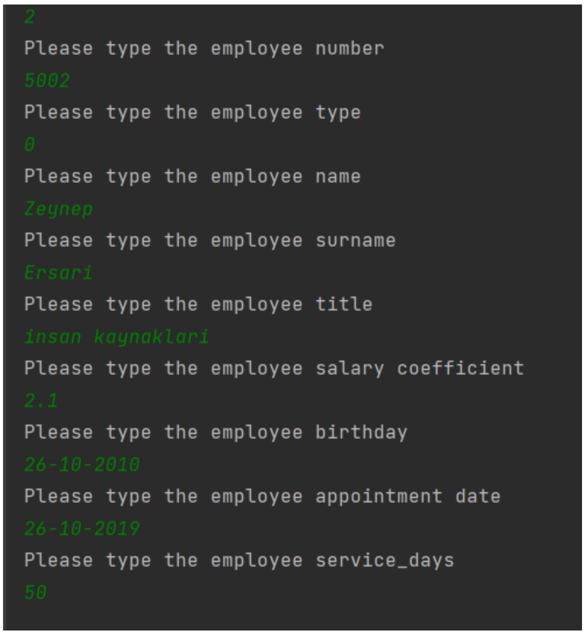
Command1



```
1
Please type the employee number
5001
Please type the employee type
1
Please type the employee name
Berat
Please type the employee surname
Ersari
Please type the employee title
Muhendis
Please type the employee salary coefficient
4.3
Please type the employee birthday
29-04-2001
Please type the employee appointment date
29-04-2020
```

If you want to add an employee who started working first time, you can use this command. When you press one, you will enter the informations one by one. Number must be 4 digits. If you want to add PermanentEmployee, employee type must be 1. For permanent employee, it is 0. Then you have to enter name, surname and title one by one. Then, you will enter the salary coefficient, it can be float or number. Then you will enter Birthday and AppointmentDate. Its format should be as above.

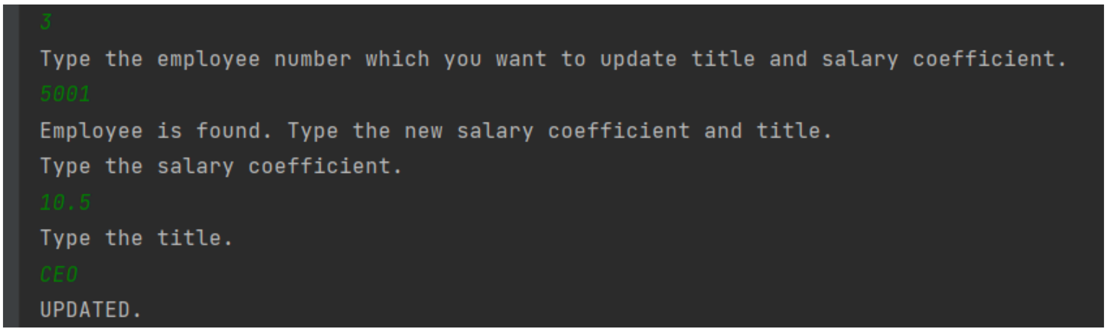
Command2



```
2
Please type the employee number
5002
Please type the employee type
0
Please type the employee name
Zeynep
Please type the employee surname
Ersari
Please type the employee title
insan kaynaklari
Please type the employee salary coefficient
2.1
Please type the employee birthday
26-10-2010
Please type the employee appointment date
26-10-2019
Please type the employee service_days
50
```

If you want to add an employee who transferred from another instution, you can use this command . When you press two, you will enter the informations one by one. Number must be 4 digits. If you want to add PermanentEmployee, employee type must be 1. For permanent employee, it is 0. Then you have to enter name, surname and title one by one. Then, you will enter the salary coefficient, it can be float or number. Then you will enter Birthday and AppointmentDate. Its format should be as above. In addition to command1, you have to enter service_days as number.

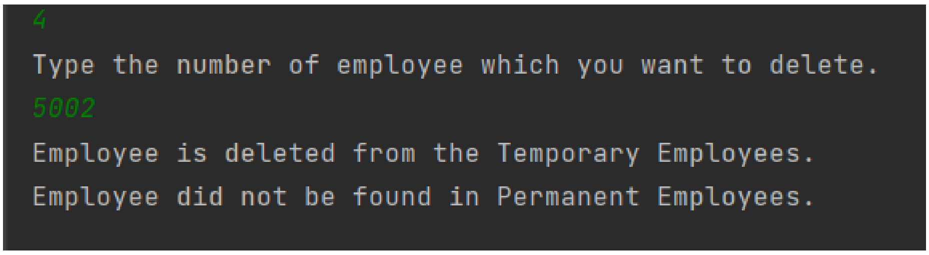
Command3



```
3
Type the employee number which you want to update title and salary coefficient.
5001
Employee is found. Type the new salary coefficient and title.
Type the salary coefficient.
10.5
Type the title.
CEO
UPDATED.
```

If you want to update salary coefficient and title, you can use this command. When you press three, you first enter a employee number. If it exists, you will enter salary coefficient and title one by one. Then, they are uptadated.

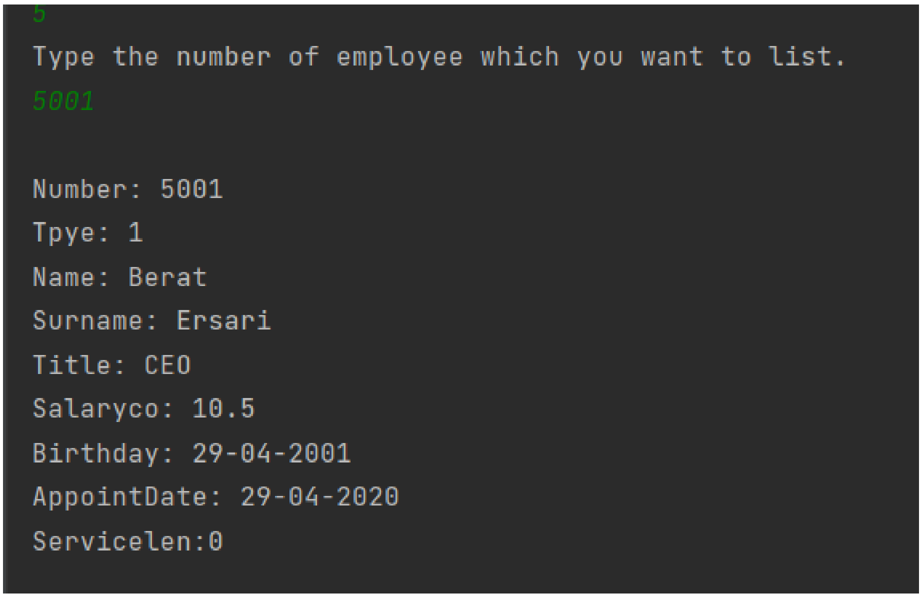
Command4



```
4
Type the number of employee which you want to delete.
5002
Employee is deleted from the Temporary Employees.
Employee did not be found in Permanent Employees.
```

If you want to delete an employee, you can use this command. When you press four, you first enter a employee number. If it exists, it will be deleted.

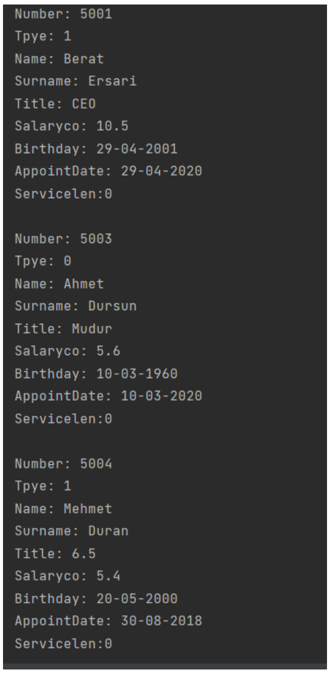
Command5



```
5
Type the number of employee which you want to list.
5001

Number: 5001
Tpye: 1
Name: Berat
Surname: Ersari
Title: CEO
Salaryco: 10.5
Birthday: 29-04-2001
AppointDate: 29-04-2020
Servicelen:0
```

If you want to display information of an employee, you can use this command. When you press five, you first enter a employee number. If it exists, its information will be displayed.

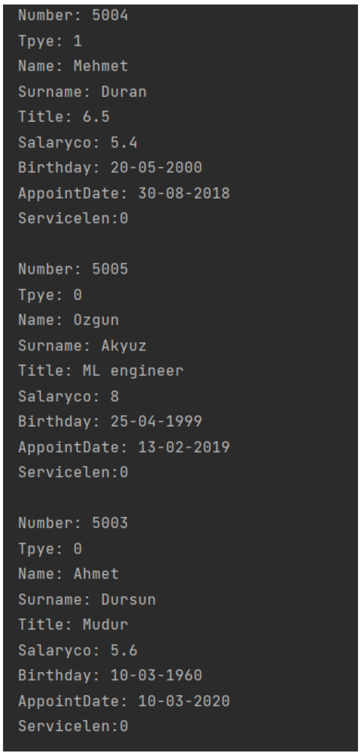
Command6



```
Number: 5001
Tpye: 1
Name: Berat
Surname: Ersari
Title: CEO
Salaryco: 10.5
Birthday: 29-04-2001
AppointDate: 29-04-2020
Servicelen:0

Number: 5003
Tpye: 0
Name: Ahmet
Surname: Dursun
Title: Mudur
Salaryco: 5.6
Birthday: 10-03-1960
AppointDate: 10-03-2020
Servicelen:0

Number: 5004
Tpye: 1
Name: Mehmet
Surname: Duran
Title: 6.5
Salaryco: 5.4
Birthday: 20-05-2000
AppointDate: 30-08-2018
Servicelen:0
```

If you want to see all employees ordered by employee number, you can use this command. When you press six, program will list all employees ordered by employee number.

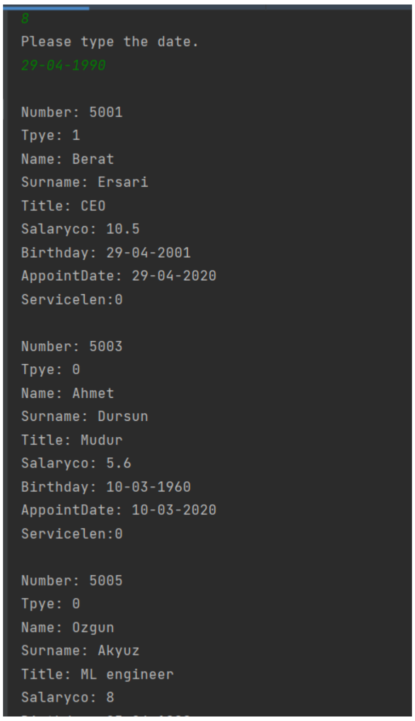
Command7



```
Number: 5004
Tpye: 1
Name: Mehmet
Surname: Duran
Title: 6.5
Salaryco: 5.4
Birthday: 20-05-2000
AppointDate: 30-08-2018
Servicelen:0

Number: 5005
Tpye: 0
Name: Ozgun
Surname: Akyuz
Title: ML engineer
Salaryco: 8
Birthday: 25-04-1999
AppointDate: 13-02-2019
Servicelen:0

Number: 5003
Tpye: 0
Name: Ahmet
Surname: Dursun
Title: Mudur
Salaryco: 5.6
Birthday: 10-03-1960
AppointDate: 10-03-2020
Servicelen:0
```

If you want to see all employees ordered by appoinment date, you can use this command. When you press seven, you program will list all employees ordered by appointment date.

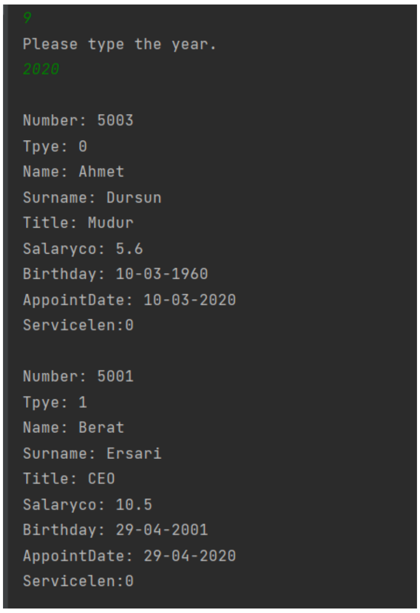
Command8



```
8
Please type the date.
29-04-1990

Number: 5001
Tpye: 1
Name: Berat
Surname: Ersari
Title: CEO
Salaryco: 10.5
Birthday: 29-04-2001
AppointDate: 29-04-2020
Servicelen:0

Number: 5003
Tpye: 0
Name: Ahmet
Surname: Dursun
Title: Mudur
Salaryco: 5.6
Birthday: 10-03-1960
AppointDate: 10-03-2020
Servicelen:0

Number: 5005
Tpye: 0
Name: Ozgun
Surname: Akyuz
Title: ML engineer
Salaryco: 8
```

If you want to see employees after a certain appointment date, you can use this command. When you press eight, you first enter a Date. Then, program will list the employees which is appointment date after this date, ordered by newest to old.

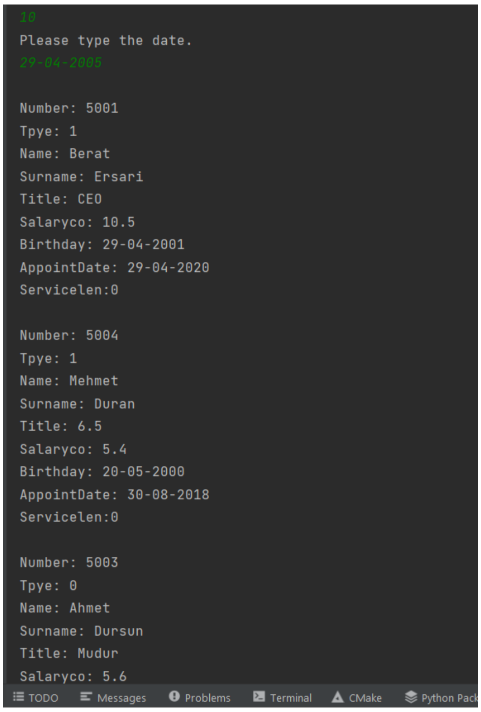
Command9



```
9
Please type the year.
2020

Number: 5003
Tpye: 0
Name: Ahmet
Surname: Dursun
Title: Mudur
Salaryco: 5.6
Birthday: 10-03-1960
AppointDate: 10-03-2020
Servicelen:0

Number: 5001
Tpye: 1
Name: Berat
Surname: Ersari
Title: CEO
Salaryco: 10.5
Birthday: 29-04-2001
AppointDate: 29-04-2020
Servicelen:0
```

If you want to see employees at a certain appointment year, you can use this command. When you press nine, you first enter a year. If it exists, program will list the employees which has appointment year with same entered year.

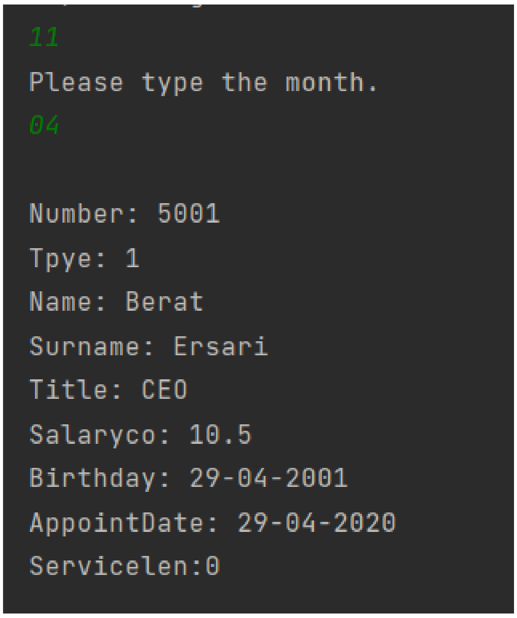
Command10



```
10
Please type the date.
29-04-2005

Number: 5001
Tpye: 1
Name: Berat
Surname: Ersari
Title: CEO
Salaryco: 10.5
Birthday: 29-04-2001
AppointDate: 29-04-2020
Servicelen:0

Number: 5004
Tpye: 1
Name: Mehmet
Surname: Duran
Title: 6.5
Salaryco: 5.4
Birthday: 20-05-2000
AppointDate: 30-08-2018
Servicelen:0

Number: 5003
Tpye: 0
Name: Ahmet
Surname: Dursun
Title: Mudur
Salaryco: 5.6
```

If you want to see employees who are born before a certain date, you can use this command. When you press ten, you first enter Date. If it exists, program will list employees who born before this date in ascording order according to employee number.

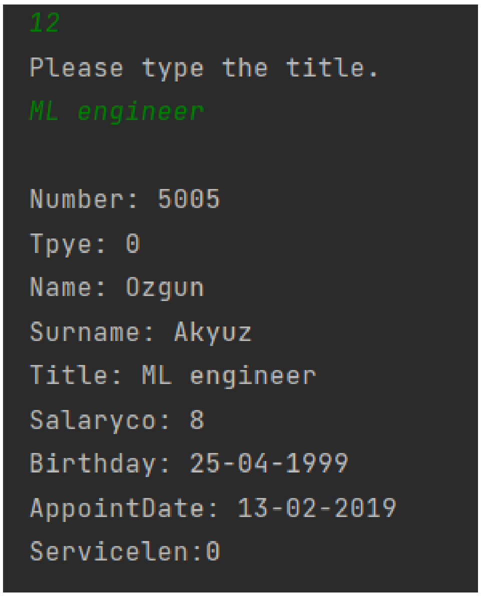
Command11



```
11
Please type the month.
04


Number: 5001
Tpye: 1
Name: Berat
Surname: Ersari
Title: CEO
Salaryco: 10.5
Birthday: 29-04-2001
AppointDate: 29-04-2020
Servicelen:0
```

If you want to see employees who born in particular month, you can use this command. When you press eleven, you first enter a month. If it exists, program will prints all employees who born in this month.

Command12



```
12
Please type the title.
ML engineer

Number: 5005
Tpye: 0
Name: Ozgun
Surname: Akyuz
Title: ML engineer
Salaryco: 8
Birthday: 25-04-1999
AppointDate: 13-02-2019
Servicelen:0
```

If you want to list information with certain a certain title, who is the last assigned to the instution, you can use this command. When you press twenty, you first enter a title. Then, program will show the last appointed employee with this title.

4- RESULT, DISCUSSION AND CONCLUSION

In my Project, I used LinkedLists and store the employees in them. These linkedlist have some methods such as print, sort, insert, delete. My results are easy to understand. Also, I think lots of the result are completely correct. I tested the code in a lot of aspects. My grading to this Project is 99. I wrote the report with care. Also, I think my code is good enough. I break one points because of efficiency of my code. Also, some parts are a bit complex.

5- REFERENCES

https://stackoverflow.com/questions/46204085/sorting-two-non-sequential-arrays-to-one-array-ascending

https://www.geeksforgeeks.org/bubble-sort-on-doubly-linked-list/