// COSC 1320 Summer 2015

// Name: Adrian

// Programming Assignment 3

// This is my own work; I will not post

#pragma once

#include <string>

#include "Administrator.h"

#include "HospitalEmployee.h"

using namespace std;

Administrator::~Administrator(void)

{

}

Administrator::Administrator() : HospitalEmployee('a' , "none" , 0){

department = "none";

}

Administrator::Administrator(char newRole, string newName, int newID, string newDept) : HospitalEmployee(newRole , newName , newID){

department = newDept;

}

void Administrator::setDept(string newDept){

department = newDept;

}

string Administrator::getDept(){

return department;

}

string Administrator::toString(){

string str = " ";

str += department;

return (HospitalEmployee::toString() + str);

}

// COSC 1320 Summer 2015

// Name: Adrian

// Programming Assignment 3

// This is my own work; I will not post

#ifndef \_\_ADMINISTRATOR\_\_

#define \_\_ADMINISTRATOR\_\_

#pragma once

#include <string>

#include "HospitalEmployee.h"

using namespace std;

class Administrator : public HospitalEmployee

{

private:

string department;

public:

Administrator();

~Administrator(void);

Administrator(char , string , int , string);

void setDept(string);

string getDept();

string toString();

};

#endif

// COSC 1320 Summer 2015

// Name: Adrian

// Programming Assignment 3

// This is my own work; I will not post

#include "Doctor.h"

#include <iostream>

using namespace std;

Doctor::Doctor(void): HospitalEmployee(){

specialty = "none";

}

Doctor::Doctor(char newRole, string newName, int newID, string newSpec) : HospitalEmployee(newRole , newName , newID){

specialty = newSpec;

//cout << "doc hit: " << specialty << endl;

}

Doctor::~Doctor(void)

{

}

string Doctor::getSpecialty(){

return specialty;

}

void Doctor::setSpecialty(string newSpec){

specialty = newSpec;

}

string Doctor::toString(){

string str = "";

str += HospitalEmployee::toString();

str += " " + specialty;

//cout << "spec: " << str << endl;

return (str);

}

// COSC 1320 Summer 2015

// Name: Adrian

// Programming Assignment 3

// This is my own work; I will not post

#ifndef \_\_DOCTOR\_\_

#define \_\_DOCTOR\_\_

#pragma once

#include "HospitalEmployee.h"

#include <string>

using namespace std;

class Doctor : public HospitalEmployee

{

private:

string specialty;

public:

Doctor();

Doctor(char , string , int , string);

~Doctor(void);

string getSpecialty();

void setSpecialty(string);

string toString();

};

#endif

// COSC 1320 Summer 2015

// Name: Adrian

// Programming Assignment 3

// This is my own work; I will not post

//#include "HospitalEmployee.h"

#include "Administrator.h"

#include "Doctor.h"

#include "EmployeeRoster.h"

#include "Janitor.h"

#include "Nurse.h"

#include "Receptionist.h"

#include "Surgeon.h"

#include <iostream>

#include <fstream>

using namespace std;

EmployeeRoster::EmployeeRoster(){

empCtr = 0;

HospitalEmployee empArr[MAX\_SIZE];

}

EmployeeRoster::~EmployeeRoster(void)

{

}

void EmployeeRoster::addEmployee(){

char newRole;

string newName;

int newID;

string newSpec;

int ptCnt;

string newDept;

char working;

cout << "Enter a role: ";

cin >> newRole;

cout << "Enter employee's name: ";

cin >> newName;

cout << "Enter employee's ID number: ";

cin >> newID;

// after getting preliminary data, it enters this flow statement to invoke the appropriate constructor

if(newRole == 'h'){

HospitalEmployee \*hosEmp = new HospitalEmployee(newRole , newName , newID);

empArr[empCtr] = hosEmp;

} else if(newRole == 'd'){

cout << "Enter the Doctor's specialty: ";

cin >> newSpec;

HospitalEmployee \*doc = new Doctor(newRole , newName , newID , newSpec);

empArr[empCtr] = doc;

(\*doc).

empCtr++;

} else if(newRole == 's'){

cout << "Enter the Surgeon's specialty: ";

cin >> newSpec;

cout << "Is the Surgeon operating? ";

cin >> working;

HospitalEmployee \*surgeon = new Surgeon(newRole , newName , newID , newSpec , working);

empArr[empCtr] = surgeon;

empCtr++;

} else if(newRole == 'n'){

cout << "Enter the Nurse's patient count: ";

cin >> ptCnt;

HospitalEmployee \*nurse = new Nurse(newRole , newName , newID , ptCnt);

empArr[empCtr] = nurse;

empCtr++;

} else if(newRole == 'a'){

cout << "Enter Administrator's department: ";

cin >> newDept;

HospitalEmployee \*admin = new Administrator(newRole , newName , newID , newDept);

empArr[empCtr] = admin;

empCtr++;

} else if(newRole == 'r'){

cout << "Enter the Receptionist's department: ";

cin >> newDept;

cout << "Is the Receptionist answering the phone? ";

cin >> working;

HospitalEmployee \*rec = new Receptionist(newRole , newName , newID , newDept , working);

empArr[empCtr] = rec;

empCtr++;

} else if(newRole == 'j'){

cout << "Enter the Janitor's department: ";

cin >> newDept;

cout << "Is the Janitor sweeping the floor? ";

cin >> working;

HospitalEmployee \*jan = new Janitor(newRole , newName , newID , newDept , working);

empArr[empCtr] = jan;

empCtr++;

} else {

cout << "Role not recognized." << endl;

}

}

void EmployeeRoster::deleteEmployee(){

string empName;

char empRole;

cout << "Enter a role: ";

cin >> empRole;

cout << "Enter a name: ";

cin >> empName;

int h = 0;

for(int i = 0; i < empCtr; i++){

if((\*empArr[i]).getName() == empName && (\*empArr[i]).getRole() == empRole){

//if there is a match, the flag (h) is triggered

h = 1;

cout << "Match found" << endl;

}

//if the flag is triggered, the array is shifted to the left by one

if(h == 1){

//delete (\*empArr[i]);

empArr[i-h] = empArr[i];

} else {

empArr[i-h] = empArr[i];

}

if(h == 0 && i == empCtr-1){

cout << "No match found." << endl;

}

cout << (\*empArr[i]).toString() << endl;

}

empCtr--;

}

void EmployeeRoster::displayEmployee(){

cout << endl;

cout << "The hospital has the following employees:" << endl;

// just loops through the array and displays the list

for(int i = 0; i < empCtr; i++){

cout << (\*empArr[i]).toString() << endl;

}

cout << endl;

}

void EmployeeRoster::readFile(){

ifstream fin;

try{

//Attempt to open the file

fin.open("Programming Assignment 3 Data.txt");

char newRole;

string newName;

int newID;

string newSpec;

int ptCnt;

char working;

string newDept;/\*\*/

while(!fin.eof()){

fin >> newRole;

fin >> newName;

fin >> newID;

if(fin.peek()){

// reads role and calls corresponding class constructor after reading appropriate data

if(newRole == 'h'){

HospitalEmployee \*hosEmp = new HospitalEmployee(newRole , newName , newID);

empArr[empCtr] = hosEmp;

empCtr++;

} else if(newRole == 'd'){

fin >> newSpec;

Doctor \*doc = new Doctor(newRole , newName , newID , newSpec);

empArr[empCtr] = doc;

cout << (\*empArr[empCtr]).toString() << endl;

empCtr++;

} else if(newRole == 'n'){

fin >> ptCnt;

Nurse \*nurse = new Nurse(newRole , newName , newID , ptCnt);

empArr[empCtr] = nurse;

empCtr++;

} else if(newRole == 's'){

fin >> newSpec;

fin >> working;

Surgeon \*surgeon = new Surgeon(newRole , newName , newID , newSpec , working);

empArr[empCtr] = surgeon;

empCtr++;

} else if(newRole == 'a'){

fin >> newDept;

Administrator \*admin = new Administrator(newRole , newName , newID , newDept);

empArr[empCtr] = admin;

empCtr++;

} else if(newRole == 'r'){

fin >> newDept;

fin >> working;

Receptionist \*rec = new Receptionist(newRole , newName , newID , newDept , working);

empArr[empCtr] = rec;

empCtr++;

} else if(newRole == 'j'){

fin >> newDept;

fin >> working;

Janitor \*jan = new Janitor(newRole , newName , newID , newDept , working);

empArr[empCtr] = jan;

empCtr++;

} else {

cout << "Role not found";/\*\*/

}

}

}

fin.close();

} catch(exception e){

//Executed if the file is not found

cout << "File not found.";

system("pause");

exit(0);

}

}

void EmployeeRoster::writeFile(){

ofstream fout;

fout.open("Programming Assignment 3 Data Out.txt");

//opens/creates the above file and writes the data to it by invoking the toString() method

for(int i = 0; i < empCtr; i++){

fout << (\*empArr[i]).toString() << endl;

}

fout.close();

}

// COSC 1320 Summer 2015

// Name: Adrian

// Programming Assignment 3

// This is my own work; I will not post

#ifndef \_\_EMPLOYEE\_ROSTER\_\_

#define \_\_EMPLOYEE\_ROSTER\_\_

#pragma once

#include "HospitalEmployee.h"

#include <string>

using namespace std;

const int MAX\_SIZE = 20;

class EmployeeRoster

{

private:

HospitalEmployee \*empArr[MAX\_SIZE];

//string empArr[MAX\_SIZE];

int empCtr;

public:

EmployeeRoster();

~EmployeeRoster(void);

void addEmployee();

void deleteEmployee();

void displayEmployee();

void readFile();

void writeFile();

};

#endif

// COSC 1320 Summer 2015

// Name: Adrian

// Programming Assignment 3

// This is my own work; I will not post

#include <iostream>

#include <sstream>

#include "HospitalEmployee.h"

#include "EmployeeRoster.h"

#include <string>

using namespace std;

HospitalEmployee::HospitalEmployee(){

role = 'h';

name = "";

idNumber = 0;

}

HospitalEmployee::HospitalEmployee(char newRole , string newName , int newID){

role = newRole;

name = newName;

idNumber = newID;

//EmployeeRoster::addEmployee(toString());

//cout << "hit: " << idNumber << endl;

}

HospitalEmployee::~HospitalEmployee(void)

{

}

char HospitalEmployee::getRole(){

return role;

}

string HospitalEmployee::getName(){

return name;

}

int HospitalEmployee::getID(){

return idNumber;

}

void HospitalEmployee::setRole(char newRole){

role = newRole;

}

void HospitalEmployee::setName(string newName){

name = newName;

}

void HospitalEmployee::setID(int newID){

idNumber = newID;

}

string HospitalEmployee::toString(){

ostringstream ss;

long num = idNumber;

ss << num;

string str = "";

str += role;

str += " ";

str += name;

str += " ";

str += ss.str();

//cout << "str: " << str << endl;

return str;

}

// COSC 1320 Summer 2015

// Name: Adrian

// Programming Assignment 3

// This is my own work; I will not post

#ifndef \_\_HOSPITALEMPLOYEE\_\_

#define \_\_HOSPITALEMPLOYEE\_\_

#pragma once

#include <string>

//#include "EmployeeRoster.h"

using namespace std;

class HospitalEmployee

{

private:

char role;

string name;

long int idNumber;

public:

HospitalEmployee(void);

HospitalEmployee(char , string , int);

~HospitalEmployee(void);

char getRole();

string getName();

int getID();

void setRole(char);

void setName(string);

void setID(int);

string toString();

};

#endif

// COSC 1320 Summer 2015

// Name: Adrian

// Programming Assignment 3

// This is my own work; I will not post

#pragma once

#include <string>

#include "Administrator.h"

#include "Janitor.h"

using namespace std;

Janitor::Janitor() : Administrator('j' , "none" , 0 , "Maintenance"){

sweeping = 'N';

}

Janitor::~Janitor(void)

{

}

Janitor::Janitor(char newRole, string newName, int newID, string newDept, char busyState):Administrator(newRole , newName , newID , newDept){

sweeping = busyState;

}

char Janitor::getSweeping(){

return sweeping;

}

void Janitor::setSweeping(char busyState){

sweeping = busyState;

}

string Janitor::toString(){

string str = " ";

str += sweeping;

return (Administrator::toString() + str);

}

// COSC 1320 Summer 2015

// Name: Adrian

// Programming Assignment 3

// This is my own work; I will not post

#ifndef \_\_JANITOR\_\_

#define \_\_JANITOR\_\_

#pragma once

#include <string>

using namespace std;

class Janitor : public Administrator

{

private:

char sweeping;

public:

Janitor(void);

~Janitor(void);

Janitor(char , string , int , string , char);

char getSweeping();

void setSweeping(char);

string toString();

};

#endif

// COSC 1320 Summer 2015

// Name: Adrian

// Programming Assignment 3

// This is my own work; I will not post

#include <iostream>

#include <fstream>

#include "EmployeeRoster.h"

#include "View.h"

using namespace std;

int main(){

View view;

EmployeeRoster list1;

list1.readFile();

int option = 0;

while(option != 4){

cout << "Make a selection: " << endl;

view.showMenu();

cin >> option;

switch(option){

case 1:

list1.addEmployee();

break;

case 2:

list1.deleteEmployee();

break;

case 3:

list1.displayEmployee();

break;

case 4:

list1.writeFile();

break;

default:

cout << "Command not recognized." << endl;

break;

}

}

cout << "Thank you for using this system! Session will now terminate." << endl;

system("pause");

exit(0);

return 0;

}

// COSC 1320 Summer 2015

// Name: Adrian

// Programming Assignment 3

// This is my own work; I will not post

#include "Nurse.h"

#include <string>

#include <sstream>

using namespace std;

Nurse::Nurse(void)

{

}

Nurse::Nurse(char newRole , string newName , int newID , int ptCnt):HospitalEmployee(newRole , newName , newID){

numPatients = ptCnt;

}

Nurse::~Nurse(void)

{

}

int Nurse::getPtCnt(){

return numPatients;

}

void Nurse::setPtCnt(int ptCnt){

numPatients = ptCnt;

}

string Nurse::toString(){

ostringstream ss;

long num = numPatients;

ss << num;

string str = " " + ss.str();

return HospitalEmployee::toString() + str;

}

// COSC 1320 Summer 2015

// Name: Adrian

// Programming Assignment 3

// This is my own work; I will not post

#ifndef \_\_NURSE\_\_

#define \_\_NURSE\_\_

#pragma once

#include <string>

#include "HospitalEmployee.h"

using namespace std;

class Nurse : public HospitalEmployee

{

private:

int numPatients;

public:

Nurse();

Nurse(char , string , int , int);

~Nurse(void);

int getPtCnt();

void setPtCnt(int);

string toString();

};

#endif

// COSC 1320 Summer 2015

// Name: Adrian

// Programming Assignment 3

// This is my own work; I will not post

#pragma once

#include <string>

#include "Administrator.h"

#include "HospitalEmployee.h"

#include "Receptionist.h"

using namespace std;

Receptionist::~Receptionist(void)

{

}

Receptionist::Receptionist() : Administrator('r' , "none" , 0 , "Talking"){

answering = 'N';

}

Receptionist::Receptionist(char newRole, string newName, int newID, string newDept, char busyState) : Administrator(newRole , newName , newID , newDept){

answering = busyState;

}

char Receptionist::getAnswering(){

return answering;

}

void Receptionist::setAnswering(char busyState){

answering = busyState;

}

string Receptionist::toString(){

string str = " ";

str += answering;

return (Administrator::toString() + str);

}

// COSC 1320 Summer 2015

// Name: Adrian

// Programming Assignment 3

// This is my own work; I will not post

#ifndef \_\_RECEPTIONIST\_\_

#define \_\_RECEPTIONIST\_\_

#pragma once

#include <string>

#include "Administrator.h"

#include "HospitalEmployee.h"

using namespace std;

class Receptionist : public Administrator

{

private:

char answering;

public:

Receptionist();

~Receptionist(void);

Receptionist(char , string , int , string , char);

char getAnswering();

void setAnswering(char);

string toString();

};

#endif

// COSC 1320 Summer 2015

// Name: Adrian

// Programming Assignment 3

// This is my own work; I will not post

#pragma once

#include <iostream>

#include <string>

#include "HospitalEmployee.h"

#include "Surgeon.h"

using namespace std;

Surgeon::Surgeon() : Doctor('h' , "none" , 0 , "none"){

operating = 'N';

}

Surgeon::~Surgeon(void)

{

}

Surgeon::Surgeon(char newRole, string newName, int newID, string newSpec, char inOp) : Doctor(newRole, newName, newID, newSpec){

operating = inOp;

}

void Surgeon::setOpStat(char inOp){

operating = inOp;

}

char Surgeon::getOpStat(){

return operating;

}

string Surgeon::toString(){

string str = " ";

str += operating;

return (Doctor::toString() + str);

}

// COSC 1320 Summer 2015

// Name: Adrian

// Programming Assignment 3

// This is my own work; I will not post

#ifndef \_\_SURGEON\_\_

#define \_\_SURGEON\_\_

#pragma once

#include <string>

#include "HospitalEmployee.h"

#include "Doctor.h"

using namespace std;

class Surgeon : public Doctor

{

private:

char operating;

public:

Surgeon(void);

Surgeon(char , string , int , string , char);

~Surgeon(void);

void setOpStat(char);

char getOpStat();

string toString();

};

#endif

// COSC 1320 Summer 2015

// Name: Adrian

// Programming Assignment 3

// This is my own work; I will not post

#pragma once

#include <iostream>

#include <string>

#include "View.h"

using namespace std;

View::View(void)

{

}

View::~View(void)

{

}

void View::showMenu(){

const int menuLength = 4;

string menu[menuLength] = {"Add Employee" , "Delete Employee" , "Display Employees" , "Save and exit"};

cout << endl;

for(int i = 0; i < menuLength; i++){

cout << (i+1) << " " << menu[i] << endl;

}

cout << endl;

}

// COSC 1320 Summer 2015

// Name: Adrian

// Programming Assignment 3

// This is my own work; I will not post

#pragma once

#include <iostream>

using namespace std;

class View

{

public:

View(void);

~View(void);

void showMenu();

};



