Hospital Management System

Made in C++

Jakub Januszewski L12

Hospital Management System

Agenda

- What have you created?
- Did you have any problems?
- Did you learn anything new?
- Waht could me imrpoved?

1) What have you created?

Project structure

```
├── include
│   ├── content
│   │   ├── Doctor.h
│   │   ├── Patient.h
│   │   ├── Person.h
│   │   └── Ward.h
│   └── menu
│       └── Menu.h
└── src
    ├── Application.cpp
    └── content
        ├── Doctor.cpp
        ├── Patient.cpp
        ├── Person.cpp
        └── Ward.cpp
```

class Person

```
class Person{
public:
    int pesel;
    string name;
    Person(string name, int pesel){this->name=name;this->pesel=pesel;}
};
```

class Doctor

```
class Doctor : public Person{
public:
    Doctor(std::string name,int pesel) : Person(name,pesel){};
};
```

class Patient

```
class Patient : public Person{
public:
    string p_disease;
    Patient(std::string name,int pesel,string p_disease) : Person(name, pesel) {this->p_disease=p_disease;}
};
```

class Ward

```cpp
class Ward{
private:
    int capacity;
    std::string h_name;
    vector <Doctor> doctors;
    vector <Patient> patients;
public:
    Ward(int capacity_,std::string h_name_, vector <Doctor> &doctors_,vector <Patient> &patients_):
        capacity{capacity_},
        h_name{h_name_},
        doctors{doctors_},
        patients{patients_}{}
```

Function to add or remove Doctor

```
void addDoctor(std::vector <Doctor> &doctors){
    string d_name; string name; string surname; int pesel;
    cout << endl << "Name: "; cin >> name;
    cout << endl << "Surname: "; cin >> surname;
    d_name = name + " " + surname;
    cout << endl << "PESEL: "; cin >> pesel;
    Doctor d1(d_name,pesel);
    doctors.push_back(d1);
}
void removeDoctor(std::vector <Doctor> &doctors){
    int number;
    for(int i=0;i<doctors.size();i++) {
        cout << i << ": " << doctors[i].name << endl;
    }
    cout << "-----------" <<endl;
    cout << "Remove doctor with number: "; cin >> number;cout<<endl;
    try{
        cout << "Succesfully removed " << doctors[number].name <<endl;
        doctors.erase(doctors.begin()+number);
    }
    catch(int number){
        cout << "You typed: "<<number<<". Nothing happened."<<endl;
    }
}
```

Function to add or remove Patient

```cpp
void addPatient(std::vector <Patient> &patient){
    string p_name; string name; string surname; int pesel; string p_disease;
    cout << endl << "Name: "; cin >> name;
    cout << endl << "Surname: "; cin >> surname;
    p_name = name + " " + surname;
    cout << endl << "PESEL: "; cin >> pesel;
    cout << endl << "Disease: "; cin >> p_disease;
    std::getline(std::cin,p_disease);
    Patient p1(p_name,pesel,p_disease);
    patient.push_back(p1);
}
void removePatient(std::vector <Patient> &patients){
    int number;
    for(int i=0;i<patients.size();i++) {
        cout << i << ": " << patients[i].name << endl;
    }
    cout << "-----------" <<endl;
    cout << "Remove patient with number: "; cin >> number;cout<<endl;
    try {
        patients.erase(patients.begin() + number);
    }
    catch(int number){
        cout << "You typed: "<<number<<". Nothing happened."<<endl;
    }
}
```

Function to display all the information

```cpp
void info(){
    cout << "Ward name: "<< h_name << endl;
    cout << "Doctors: ";
    for(Doctor &d : doctors){
        cout<<d.name<<", ";
    }
    cout << "\nPatients: ";
    for(Patient &p : patients){
        cout<<p.name<<", ";
    }
    cout<<endl<<endl;
}
void personel(){
    cout << "Doctors: " << doctors.size()<<endl;
    cout << "Patients: " << patients.size()<<endl;
    cout << h_name << " capacity: " << capacity << endl;
    cout << "Free seats: " << capacity - doctors.size() - patients.size() << endl;
}
```

User Interface

```cpp
class Menu {
    vector<string> options;
public:
    Menu(vector<string> options){this->options=options;}
    int display(){
        int option;
        cout<<"--- MENU ---"<<endl;
        for(int i = 0; i<options.size(); i++){
            cout<<i<<": "<<options[i]<<endl;
        }
        cout<<options.size()<<": "<<"Exit"<<endl;
        cout<<"Your option: ";
        cin>>option;
        cout<<endl;
        return option;
    }
};
```

Menu usage in example

```
vector <string> options = {"Add patient to orthodontia","Add patient to stomatology","Add patient to physiotherapy"};
Menu m1(options);
int choice_p_ward = m1.display();
switch (choice_p_ward){
  case 0:
    addPatient(p_orthodontia);
    break;
  case 1:
    addPatient(p_stomatology);
    break;
  case 2:
    addPatient(p_physiotherapy);
    break;
  default:
    break;
}
```

2) Did you have any problems?

Yes, I had some problems, especially with class Ward:

Had to really think how to display all the doctor and patient names in void info().

```cpp
void info(){
        cout << "Ward name: "<< h_name << endl;
        cout << "Doctors: ";
        for(Doctor &d : doctors){
            cout<<d.name<<", ";
        }
        cout << "\nPatients: ";
        for(Patient &p : patients){
            cout<<p.name<<", ";
        }
        cout<<endl<<endl;
}
```

3) Did you learn anything new?

Yes, I learned how to use Slidev

And also I learned how to work with std::vector

```cpp
void removePatient(std::vector <Patient> &patients){
    int number;
    for(int i=0;i<patients.size();i++) {
        cout << i << ": " << patients[i].name << endl;
    }
    cout << "-----------" <<endl;
    cout << "Remove patient with number: "; cin >> number;cout<<endl;
    try {
        patients.erase(patients.begin() + number);
    }
    catch(int number){
        cout << "You typed: "<<number<<". Nothing happened."<<endl;
    }
}
```

4) What could be improved?

Hospital Management System

What could be improved?

- Doctor can be asigned to specific patients, not just Ward.
- More methods and constructors.
- class Menu can have vector of strings and vector of methods.
- switch statement can be used as an OOP method.

Thanks for your attention :)