

Hospital

Приложение 1

Оглавление

1	Алфавитный указатель классов	1
1.1	Классы	1
2	Классы	3
2.1	Класс Button	3
2.1.1	Подробное описание	3
2.1.2	Методы	3
2.1.2.1	CheckEvent()	4
2.1.2.2	DisselectButton()	5
2.1.2.3	GetY()	5
2.1.2.4	SelectButton()	5
2.2	Класс Department	6
2.2.1	Подробное описание	6
2.2.2	Конструктор(ы)	6
2.2.2.1	Department()	6
2.3	Класс File	7
2.3.1	Подробное описание	7
2.3.2	Методы	7
2.3.2.1	ReadDepartment()	7
2.3.2.2	ReadPatient()	8
2.3.2.3	WriteDepartment()	8
2.3.2.4	WritePatient()	8
2.4	Класс Menu	9
2.4.1	Подробное описание	10
2.4.2	Методы	10
2.4.2.1	finalGame()	10
2.4.2.2	FreePatient()	10
2.4.2.3	getRandomNumber()	10
2.4.2.4	StartGame()	11
2.5	Класс Patient	11
2.5.1	Подробное описание	12
2.5.2	Конструктор(ы)	12
2.5.2.1	Patient()	12
	Алфавитный указатель	13

Глава 1

Алфавитный указатель классов

1.1 Классы

Классы с их кратким описанием.

Button	Класс Кнопок	3
Department	Класс Отделение больницы	6
File	Класс для работы с файлами базы данных	7
Menu	Класс Работы с базами данных	9
Patient	Класс Пациента	11

Глава 2

Классы

2.1 Класс Button

Класс Кнопок

```
#include <button.h>
```

Открытые члены

- `Button ()`
Конструктор
- `void SelectButton (int a)`
Создает выделенную кнопку
- `void DisselectButton (int a)`
Создает невыделенную кнопку
- `void PrintButtons ()`
Рисует на экране все кнопки
- `void UpCursor ()`
Перемещение курсора на позицию вверх
- `void DownCursor ()`
Перемещение курсора на позицию вниз
- `int CheckEvent (int n)`
Проверка события, сделанного пользователем
- `int GetY ()`
Положение курсора

2.1.1 Подробное описание

Класс Кнопок

Создает кнопки Отрисовывает кнопки на экране Проверяет события взаимодействия с клавиатурой

2.1.2 Методы

2.1.2.1 CheckEvent()

```
int Button::CheckEvent (  
    int n )
```

Проверка события, сделанного пользователем

Аргументы

in	n	нажатая клавиша на клавиатуре пользователем
----	---	---

2.1.2.2 DisselectButton()

```
void Button::DisselectButton (  
    int a )
```

Создает невыделенную кнопку

Аргументы

in	a	номер название кнопки в массиве имен
----	---	--------------------------------------

Кнопка является невыделенной, если позиция курсора не соответствует позиции кнопки

2.1.2.3 GetY()

```
int Button::GetY ( )
```

Положение курсора

Возвращает

у целое число положение курсора

2.1.2.4 SelectButton()

```
void Button::SelectButton (  
    int a )
```

Создает выделенную кнопку

Аргументы

in	a	номер название кнопки в массиве имен
----	---	--------------------------------------

кнопка является выделенной, если позиция курсора соответствует позиции кнопки

Объявления и описания членов классов находятся в файлах:

- button.h
- button.cpp

2.2 Класс Department

Класс Отделение больницы

```
#include <department.h>
```

Открытые члены

- [Department](#) ()
конструктор
- [Department](#) (string nm, string dis[2], int freep, int numberp, int ind)
Конструктор
- string GetName ()
- string * GetDiseases ()
- int GetFreePlaces ()
- int GetNumberPlaces ()
- int GetIndex ()
- void SetName (string nm)
- void SetDiseases (string dis[])
- void SetFreePlaces (int freep)
- void SetNumberPlaces (int freen)
- void SetIndex (int ind)

Друзья

- ostream & operator<< (ostream &out, const [Department](#) &exc)
- istream & operator>> (istream &in, [Department](#) &exc)

2.2.1 Подробное описание

Класс Отделение больницы

2.2.2 Конструктор(ы)

2.2.2.1 Department()

```
Department::Department (
    string nm,
    string dis[2],
    int freep,
    int numberp,
    int ind )
```

Конструктор

Аргументы

in	nm	строка: название
in	dis	массив строк: болезни
in	fplace	целое число: количество свободных мест
in	nplace	целое число: количество мест
in	ind	целое число: номер отделения

Объявления и описания членов классов находятся в файлах:

- department.h
- department.cpp

2.3 Класс File

Класс для работы с файлами базы данных

```
#include <file.h>
```

Открытые члены

- `File ()`
Конструктор
- `vector< Patient > * ReadPatient (char *f, vector< Patient > *p)`
Чтение данных пациентов из файла
- `vector< Department > * ReadDepartment (char *f, vector< Department > *d)`
Чтение данных отделений больницы из файла
- `void WriteDepartment (char *f, vector< Department > d)`
Запись данных отделений в файл
- `void WritePatient (char *f, vector< Patient > p)`
Запись данных пациентов
- `void CheckFiles ()`
Проверка наличия и целостности всех файлов

2.3.1 Подробное описание

Класс для работы с файлами базы данных

2.3.2 Методы

2.3.2.1 ReadDepartment()

```
vector< Department > * File::ReadDepartment (
    char * f,
    vector< Department > * d )
```

Чтение данных отделений больницы из файла

Аргументы

in	*f	указатель на файл
in	*p	указатель на массив отделений

Возвращает

*d указатель на массив Отделений

2.3.2.2 ReadPatient()

```
vector< Patient > * File::ReadPatient (
    char * f,
    vector< Patient > * p )
```

Чтение данных пациентов из файла

Аргументы

in	*f	указатель на файл
in	*p	указатель на массив пациентов

Возвращает

vector<Patient> *p указатель на массив пациентов

2.3.2.3 WriteDepartment()

```
void File::WriteDepartment (
    char * f,
    vector< Department > d )
```

Запись данных отделений в файл

Аргументы

in	*f	указатель на файл
in	*p	указатель на массив отделений

2.3.2.4 WritePatient()

```
void File::WritePatient (
```

```
char * f,
vector< Patient > p )
```

Запись данных пациентов

Аргументы

in	*f	указатель на файл
in	*p	указатель на массив пациентов

Объявления и описания членов классов находятся в файлах:

- file.h
- file.cpp

2.4 Класс Menu

Класс Работы с базами данных

```
#include <menu.h>
```

Открытые члены

- `Menu ()`
Конструктор
- `int GetTime ()`
- `int GetHealen ()`
- `int GetDied ()`
- `void PrintPatient ()`
Выводит на экран всех пациентов в больницы
- `void PrintDepartment ()`
Выводит на экран все отделения больницы
- `void AddPatient ()`
Добавляет нового пациента в больницу
- `void AddDepartment ()`
Добавляет новое отделение в больницу
- `void DaySkip ()`
Смещение по времени
- `void HospitalAdmission ()`
Добавление пациента в больницу
- `void StartGame (char *argv[])`
Начало игры
- `void finalGame (char *argv[])`
Конец игры
- `void FreePatient ()`
Выписка пациента из больницы
- `int getRandomNumber (int min, int max)`
Получение рандомного числа в диапазоне
- `int getch (void)`
функция ввода в терминал без echo.

2.4.1 Подробное описание

Класс Работы с базами данных

2.4.2 Методы

2.4.2.1 finalGame()

```
void Menu::finalGame (
    char * argv[] )
```

Конец игры

Аргументы

in	*argv[]	указатель на массив с переданными файлами
----	---------	---

Запись базы данных в файлы

2.4.2.2 FreePatient()

```
void Menu::FreePatient ( )
```

Выписка пациента из больницы

Проверка пациентов на статус, если пациент имеет статус выписан, то подлежит выписке если пациент имеет статус умер, то подлежит переводу в морг

2.4.2.3 getRandomNumber()

```
int Menu::getRandomNumber (
    int min,
    int max )
```

Получение рандомного числа в диапазоне

Аргументы

in	min	- минимальное число в диапазоне
in	max	- максимальное число в диапазоне

Возвращает

целое рандомное число в диапазоне между min и max

2.4.2.4 StartGame()

```
void Menu::StartGame (
    char * argv[] )
```

Начало игры

Аргументы

in	*argv[]	указатель на массив с переданными файлами
----	---------	---

Печать логотипа, стартовое меню игры Проверка целостности файлов игры Чтение базы данных из файла

Объявления и описания членов классов находятся в файлах:

- menu.h
- menu.cpp

2.5 Класс Patient

Класс Пациента

```
#include <patient.h>
```

Открытые члены

- [Patient](#) ()
Конструктор
- [Patient](#) (string fname, string lname, string dis, string type_dis, int st, int dep, int timet, int timel)
Конструктор
- string GetFirstName ()
- string GetLastName ()
- string GetDiagnosis ()
- string GetTypeOfDisease ()
- int GetStatus ()
- int GetDepartment ()
- int GetTimeTreat ()
- int GetTimeLeft ()
- void SetFirstName (string fname)
- void SetLastName (string lname)
- void SetDiagnosis (string dis)
- void SetTypeOfDisease (string type_dis)
- void SetStatus (int st)
- void SetDepartment (int dep)
- void SetTimeTreat (int timet)
- void SetTimeLeft (int timel)

Друзья

- ostream & operator<< (ostream &out, const Patient &exc)
- istream & operator>> (istream &in, Patient &exc)

2.5.1 Подробное описание

Класс Пациента

2.5.2 Конструктор(ы)

2.5.2.1 Patient()

```
Patient::Patient (
    string fname,
    string lname,
    string dis,
    string type_dis,
    int st,
    int dep,
    int timet,
    int timel )
```

Конструктор

Аргументы

in	fname	строка: имя
in	lname	строка: фамилия
in	dis	строка: диагноз
in	type_dis	строка: тип диагноза по типу травмы
in	st	целое число: статус пациента
in	dep	целое число: номер отделение больницы
in	timet	целое число: отведенное время на лечение
in	timel	целое число: время до смерти

Объявления и описания членов классов находятся в файлах:

- patient.h
- patient.cpp

Предметный указатель

- Button, [3](#)
 - CheckEvent, [3](#)
 - DisselectButton, [5](#)
 - GetY, [5](#)
 - SelectButton, [5](#)
- CheckEvent
 - Button, [3](#)
- Department, [6](#)
 - Department, [6](#)
- DisselectButton
 - Button, [5](#)
- File, [7](#)
 - ReadDepartment, [7](#)
 - ReadPatient, [8](#)
 - WriteDepartment, [8](#)
 - WritePatient, [8](#)
- finalGame
 - Menu, [10](#)
- FreePatient
 - Menu, [10](#)
- getRandomNumber
 - Menu, [10](#)
- GetY
 - Button, [5](#)
- Menu, [9](#)
 - finalGame, [10](#)
 - FreePatient, [10](#)
 - getRandomNumber, [10](#)
 - StartGame, [10](#)
- Patient, [11](#)
 - Patient, [12](#)
- ReadDepartment
 - File, [7](#)
- ReadPatient
 - File, [8](#)
- SelectButton
 - Button, [5](#)
- StartGame
 - Menu, [10](#)
- WriteDepartment
 - File, [8](#)
- WritePatient
 - File, [8](#)