// Базовый класс "Судебное заседание"

class Hearing

{

// Структура "Дата судебного заседания"

struct Atrial\_Date

{

int \_day;//день

int \_month;//месяц

int \_year;//год

};

//Структура "Предмет судебного процесса"

struct Subject\_of\_a\_Court\_Case

{

int \_id\_number\_of\_a\_court\_case;// идентификационный номер судебного процесса

string \_name\_of\_a\_court\_case;//название судебного процесса

int \_members\_number\_of\_a\_court\_case;//число участников судебного процесса

};

//Базовый абстрактный класс "Участник судебного процесса"

class Member\_of\_a\_Court\_Case

{

bool \_member\_ready\_in\_the\_court\_case;// булева переменная показывает , присутствует ли участник судебного процесса на суде

string \_name\_surname\_member\_court\_case;//Ф.И.О. участника судебного процесса

//Структура "Дата рождения участника судебного процесса"

struct Birth\_Date\_Member\_Court\_Case

{

int \_birth\_day;

int \_birth\_month;

int \_birth\_year;

};

string \_place\_birth\_member\_court\_case;//Место рождения участника судебного процесса

//Структура "Адрес прописки участника судебного процесса"

struct Adress\_Member\_Court\_Case

{

string \_town\_living;//город проживания

string \_district\_living;//район проживания

string \_street\_living;//улица проживания

string \_house\_living;//дом

string \_flat\_living;//квартира

};

public:

// Объявление конструктора абстрактного базового класса "Участник судебного процесса"

Member\_of\_a\_Court\_Case(bool member\_IsReady = true);

//Функция Наличие участника создается как чистая виртуальная функция для последующего переопределения

void virtual Presence\_Members\_Hearing()const = 0;

//Функция "Персональные данные члена судебного заседания" создается как виртуальная функция для последующего переопре*делен*ия

void virtual Getting\_Information\_Members\_Hearing()const;

//Реализация конструктора абстрактного базового класса "Участник судебного процесса"

Member\_of\_a\_Court\_Case::Member\_of\_a\_Court\_Case(bool membrer\_IsReady) : \_member\_ready\_in\_the\_court\_case(member\_IsReady){}

void Member\_of\_a\_Court\_Case::Presence\_Members\_Hearing()const

{

if (\_member\_ready\_in\_the\_court\_case)

cout << "Все участники судебного заседания в сборе ,судебное заседание состоится. " << endl;

else

cout << "При отсутствии хотя бы одного участника судебное заседание не состоится." << endl;

}

};

};

//Объявление производного класса Plaintiff(Истец)

class Plaintiff :public Member\_of\_a\_Court\_Case

{

void Plaintiff::Presence\_Members\_Hearing()const

{

if (Plaintiff:\_member\_ready\_in\_the\_court\_case)

cout << "Истец присутствует. " << endl;

else

cout << "Истец отсутствует." << endl;

}

};

//Объявление производного класса Lower(Адвокат)

class Lower :public Member\_of\_a\_Court\_Case

{

void Lower::Presence\_Members\_Hearing()const

{

if (Lower:\_member\_ready\_in\_the\_court\_case)

cout << "Адвокат присутствует. " << endl;

else

cout << "Адвокат отсутствует." << endl;

}

};

//Объявление производного класса Judge

class Judge :public Member\_of\_a\_Court\_Case

{

void Judge::Presence\_Members\_Hearing()const

{

if (Judge:\_member\_ready\_in\_the\_court\_case)

cout << "Судья присутствует. " << endl;

else

cout << "Судья отсутствует." << endl;

}

};

//Объявление производного класса Defendant(Ответчик)

class Defendant :public Member\_of\_a\_Court\_Case

{

void Defendant::Presence\_Members\_Hearing()const

{

if (Defendant:\_member\_ready\_in\_the\_court\_case)

cout << "Ответчик присутствует. " << endl;

else

cout << "Ответчик отсутствует." << endl;

}

};

//Объявление производного класса Prosecutor(Обвинитель/Прокурор)

class Prosecutor :public Member\_of\_a\_Court\_Case

{

void Prosecutor::Presence\_Members\_Hearing()const

{

if (Prosecutor:\_member\_ready\_in\_the\_court\_case)

cout << "Обвинитель присутствует. " << endl;

else

cout << "Обвинитель отсутствует." << endl;

}

};

int main()

{

//Создание объекта класса "Истец"

Plaintiff aPlaintiff;

aPlaintiff.Presence\_Members\_Hearing();

return 0;

}