

## COI 管理系统 C++ API 说明

API 提供的功能如下：

- 一、新建，更新，查询病人个人信息
- 二、新建，更新，查询患者某次住院信息
- 三、查询某个患者的历史住院信息
- 四、查询或者更新所有状态是住院中的患者信息

有以下两个结构体

```
struct Person_info
{
    int id;
    char* name;
    char sex;
    int age;
    char* id_card_num;
    char* mar_stu;
    char* edu_stu;
    char* photo_addr;
};
```

Person\_info 结构体说明

项目	数据类型	说明
id	int	患者在数据库中的唯一标示符，在往数据库中写入数据时，任何时候都不能提供该值，此数据用数据库自动维护。该值仅用于查询。
name	长度 20 以内的字符串	患者姓名
sex	F（女），M（男）	性别
age	INT	年龄
Id_card_num	长度固定 18，字符串	身份证号
mar_stu	建议长度为 2，字符串	婚姻状况
edu_su	建议长度为 2，字符串	教育信息
photo_addr	患者头像的绝对路径	存放患者头像照片地址

```
struct Medical_info
{
    int person_id;
    int medical_num;
    char dignose;
    char* place;
    int danger_level;
```

```

char* danger_desc;
char cure_meth;
double in_panss;
double out_panss;
char* wtgo;
}

```

Medical\_info 结构体说明

项目	数据类型	说明
person_id	int	该条住院信息所属的患者的个人信息位于数据库中的 ID
medical_num	Int	患者住院病号
dignose	Char	一个字母表示诊断类型
place	长度为 50 以内的字符串	事发地点
danger_level	Int	危险等级
danger_desc	长度为 50 以内的字符串	危险行为描述
cure_meth	Char	一个字母代表的治疗方案
in_panss	double	入院 panss
out_panse	double	出院 panss
wtgo	长度为 50 以内的字符串	患者去向

一：新建，更新，删除，查询患者个人信息

1. 新建个人信息

函数 int create\_person\_info(Person\_info person)

参数说明：

在数据库中新建一个个人信息，并插入 person 中封装的所有信息

返回值说明：

返回新插入的患者 ID，位于数据库中的唯一数字标号，返回 -1 表示插入失败。

2. 更新个人信息

函数 bool update\_person\_info(int id, Person\_info person)

参数说明：

参数	数据类型	说明
id	int(NN)	患者在数据库中的 ID

person 中封装了需要更新的新信息，空值表示不更新该项。

3. 查询

函数： Person\_info\* get\_person\_by\_id(int id);

参数说明： 需要查询的患者的 id;

返回值说明： 若查询成功，返回一个 `Person_info*`，里面封装了查询到的信息，若查询失败，则返回 `NULL`。

## 五、新建，更新，查询患者某次住院信息

### 1. 新建患者 住院信息

函数 `bool create_medical_info(int person_id, Medical_info info);`

参数说明： `person_id` 为患者的 id，用 `info` 封装的信息新建一条住院记录。

返回值： 返回 `true` 表示数据新建成功，返回 `false` 表示数据新建失败

### 2. 更新患者 住院信息

函数 `bool update_medical_info(int person_id, Medical_info info);`

参数说明： `person_id` 为患者的 id，用 `info` 封装的信息更新一条住院记录，空值表示保原值不更新。

返回值: `true` 表示更新成功，`false` 表示更新失败

### 3. 查询患者 住院信息

函数 `Medical_info* get_medical_info(int person_id);`

参数 `person_id` 需要查询的患者在数据库中的 ID

返回值： 返回一个 `Medical_info` 结构体指针，指向里面封装了该患者最近一次的住院的信息，返回 `NULL` 表示数据库中不存在记录。

## 三. 查询某个患者的历史住院信息

### 1. 函数 `vector<Medical_info> get_history_medical_info(int person_id)`

参数 `person_id` 表示需要查询的患者 id，

返回值 一个 `vector<Medical_info>` 动态数组，里面封装了该患者的所有住院信息，  
查询失败返回 `vector<Person_info>` 的 `size()` 查询为 0

### 2. 函数 `vector<Medical_info> get_period_medical_info(int person_id, char* start_date, char* end_date);`

参数 `person_id` 表示需要查询的患者位于数据库中的 ID。

`start_date` 表示需要查询从 `start_date` 开始的住院信息，可传空值 `NULL`

`end_date` 表示需要查询位于 `end_date` 以前的住院信息，可传空值 `NULL`

时间字符串格式 `yyyy-mm-dd`

缺省值处理： 若 `start_date` 缺省，则查询 `end_date` 以前的所有数据

若 `end_date` 缺省，则查询从 `start_date` 开始到现在的所有数据

若同时缺省，则查询所有历史住院数据

参数约束： 若 `start_date` 与 `end_date` 同时不为空，若 `start_date` 与 `end_date` 表示的时间区间为空（`start_date` 大于 `end_date`），那么返回一个长度为 0 的 `vector<Medcal_info>`。

返回值 返回储存了查询到的所有数据，封装在 `vector<Medical_info>` 中，

查询失败返回 `vector<Medical_info>` 的 `size()` 查询为 0

四. 查询所有状态是住院中的患者信息

函数 `vector<Person_info> get_person_by_medical_stu();`

参数 无

返回值 返回一个封装所有正在住院的所有患者的个人信息的 `vector<Person_info>`。

查询失败返回 `vector<Person_info>` 的 `size()` 查询为 0

备注:

类型空值表示

类型	空值表示
INT（整型）	-1
char（字符串）	NULL

患者住院信息表中的 医疗状态定义

定义	数值	表示
STATU_IN	1	住院中
STATU_OUT	2	已出院