



**数据结构课程设计**

**<数据结构说明报告>**

**兰学超 2019211564**

**李培阳 2019211580**

**熊瑞东 2018211612**

# 数据结构说明

注：省略了所有的构造函数、析构函数、setter（set函数）与所有的getter（get函数）

# 1 Vertex——顶点

**（1）数据结构定义**

class Vertex

{

private:

    int no;

    qreal x;

    qreal y;

    QString description;

    bool isSpot;

};

**（2）成员变量意义**

int no：顶点的编号；

qreal x：该顶点的x坐标；

qreal y：该顶点的y坐标；

QString description：该顶点的相关描述信息；

bool isSpot：该点是否为有实际意义的节点的标志。

# 2 Edge——有向边

**（1）数据结构定义**

class Edge

{

private:

    qreal distance;

    qreal crowdness;

    qreal velocity;

};

**（2）成员变量意义**

qreal distance：该边的总距离

qreal crowdness：该边的拥挤度

qreal velocity：该边可允许通行的最大速度

# 3 current——当前状态

**（1）数据结构定义**

class Current

{

public:

    bool inShahe();

    bool inBenbu();

    bool inBetween();

private:

    qreal x;

    qreal y;

    qreal velocity;

    Priority priority;

    QVector<QMap<int,Edge\*>> adj\_map;

    QVector<Vertex\*> vertices;

    QVector<Vertex\*> path;

    Vertex\* last\_ver;

    int next\_ver\_index;

    qreal total\_dis;

    qreal current\_dis;

    qreal total\_pass;

    qreal current\_pass;

    qreal total\_time;

    qreal total\_pass\_time;

};

**（2）成员变量意义**

qreal x：当前位置的x坐标；

qreal y：当前位置的y坐标；

qreal velocity：当前所在边的允许通行速度；

Priority priority：当前的选择策略（优先度）；

QVector<QMap<int,Edge\*>> adj\_map：邻接表；

QVector<Vertex\*> vertices：顶点信息表；

QVector<Vertex\*> path：途径数组，包括起点；

Vertex\* last\_ver：上一顶点，当current\_pass==0时，last\_ver为当前节点；

int next\_ver\_index：下一个顶点在途径数组中的下标；

qreal total\_dis：路径总距离；

qreal current\_dis：当前边的长度；

qreal total\_pass：总已走距离；

qreal current\_pass：当前已走距离（距离上一个节点的距离）；

qreal total\_time：总所需时间；

qreal total\_pass\_time：总已经历的时间。

**（3） 成员函数说明**

bool inShahe()：判断当前是否在沙河校区；

bool inBenbu()：判断当前是否在本部校区；

bool inBetween()：判断当前是否在校外。

# 4 Log——日志记录

**（1）数据结构定义**

class Log

{

public:

    void readLog();

    void writeLog();

    void addItem(Clock clk, QString op, QString inf);

    void popItem();

    QVector<QString> search(Clock clk\_high, Clock clk\_low, const QString &ope) const;

private:

    typedef struct ITEM {

        Clock clk;

        QString ope;

        QString info;

    } item;

    QVector<item> log\_list;

};

**（2）成员变量意义**

typedef struct ITEM {

Clock clk：日志记录的时间；

QString ope：日志记录的操作信息；

QString info：日志记录的解释性信息；

} item：日志记录的结构体；

QVector<item> log\_list：日志记录数组，即日志记录

**（3）成员函数说明**

void readLog()：读日志文件；

void writeLog()；写日志文件；

void addItem(Clock clk, QString op, QString inf)：

向日志记录数组中写入一条记录；

void popItem()：删除日志记录数组中的最后一条记录；

QVector<QString> search(Clock clk\_high, Clock clk\_low, const QString &ope) const：

搜索一定时间范围内的所有日志记录，并格式化为字符串返回。

# 5 Account——账号

**（1）数据结构定义**

class Account

{

public:

    Course gotoclass(Clock clock) const;

private:

    QString id;

    QString password;

    QVector<Course> courses;

};

**（2）成员变量意义**

QString id：账户名；

QString password：密码；

QVector<Course> courses：课程表。

**（3）成员函数说明**

Course gotoclass(Clock clock) const：

对应当前时间找到正在进行的课程并返回。

# 6 Clock

**（1）数据结构定义**

class Clock

{

public:

    QString get\_clock\_str();

    void update\_clock();

    int difSec(const Clock &clock);

    bool operator>=(const Clock &clock);

    Clock operator-(int h);

private:

    int sec;

    int min;

    int hour;

    int wday;

};

**（2）成员变量意义**

int sec：秒；

int min：分；

int hour：时；

int wday：星期（0-6，0表示周日）。

**（3）成员函数说明**

QString get\_clock\_str()：将时钟转化为格式化字符串并返回；

void update\_clock()：时钟更新，+1s；

int difSec(const Clock &clock)：求两时钟相差的秒数；

bool operator>=(const Clock &clock)：重载运算符>=：用于判断时间的大于等于关系；

Clock operator-(int h)：重载运算符-：用于计算时间减去传入的小时数后得到的时间。

# 7 Course——课程

**（1）数据结构定义**

class Course

{

private:

    QString course\_name;

    Clock begin\_time;

    Clock end\_time;

    int classroom;

};

**（2）成员变量意义**

QString course\_name：课程名；

Clock begin\_time：课程开始时间；

Clock end\_time：课程结束时间；

int classroom：教室对应的顶点编号。

# 8 adj\_map——邻接表

**（1）数据结构定义**

QVector<QMap<int,Edge\*>> adj\_map;

1. **成员变量意义**

QVector的下标：顶点编号；

QMap的key：邻接点的序号；

QMap的value：顶点到Key的有向边；

# 9 vertices——顶点数组

**（1）数据结构定义**

QVector<Vertex\*> vertices;

# 10 logic2phy——逻辑名到顶点的映射表

**（1）数据结构定义**

QMap<QString,QVector<int>> logic2phy;

**（2）成员变量意义**

QMap的key：逻辑名；

QMap的value：对应的物理名的顶点编号数组；

QVector的值：物理名的顶点编号。