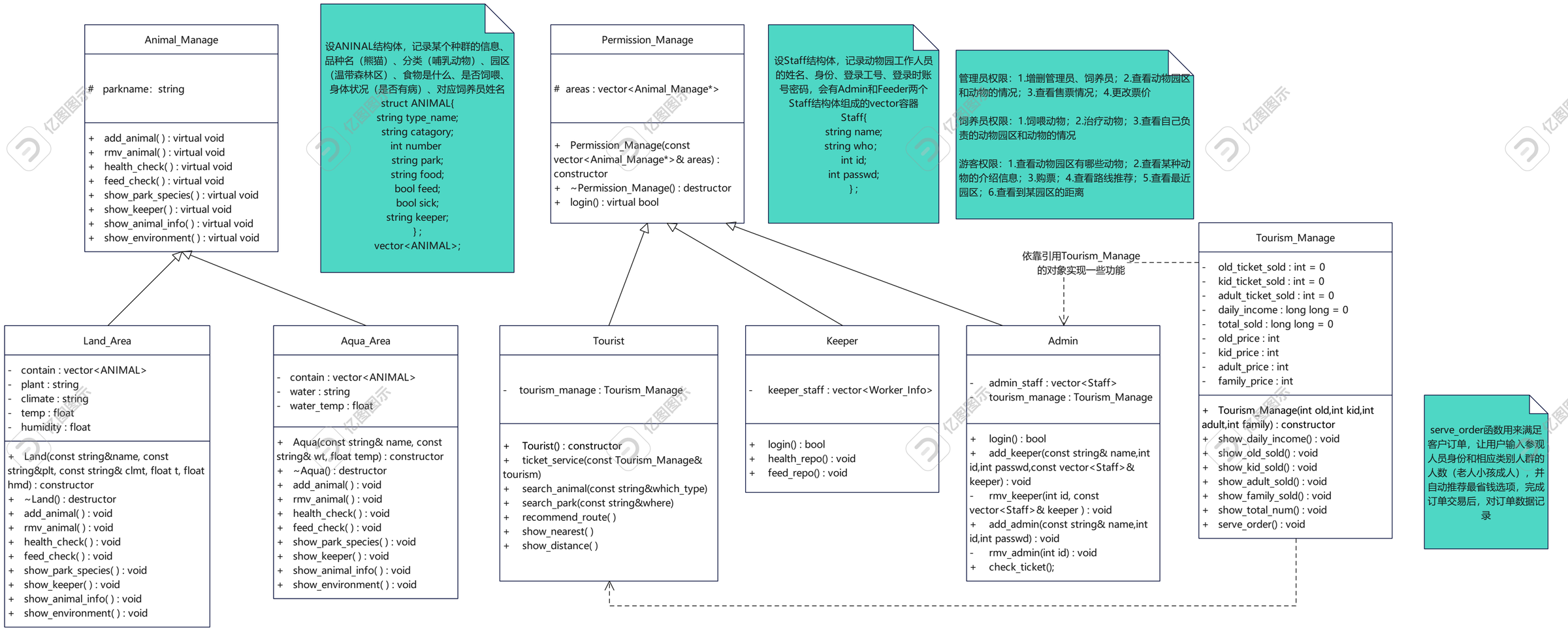
Animal_Manage
parkname: string
+ add_animal( ) : virtual void
+ rmv_animal( ) : virtual void
+ health_check( ) : virtual void
+ feed_check( ) : virtual void
+ show_park_species( ) : virtual void
+ show_keeper( ) : virtual void
+ show_animal_info( ) : virtual void
+ show_environment( ) : virtual void
设ANINAL结构体，记录某个种群的信息、品种名（熊猫）、分类（哺乳动物）、园区（温带森林区）、食物是什么、是否饲喂、身体状况（是否有病）、对应饲养员姓名
struct ANIMAL{
string type_name;
string catagory;
int number
string park;
string food;
bool feed;
bool sick;
string keeper;
} ;
vector<ANIMAL>;
Permission_Manage
areas : vector<Animal_Manage*>
+ Permission_Manage(const vector<Animal_Manage*>& areas) : constructor
+ ~Permission_Manage() : destructor
+ login() : virtual bool
设Staff结构体，记录动物园工作人员的姓名、身份、登录工号、登录时账号密码，会有Admin和Feeder两个Staff结构体组成的vector容器
Staff{
string name;
string who;
int id;
int passwd;
} ;
管理员权限：1.增删管理员、饲养员；2.查看动物园区和动物的情况；3.查看售票情况；4.更改票价
饲养员权限：1.饲喂动物；2.治疗动物；3.查看自己负责的动物园区和动物的情况
游客权限：1.查看动物园区有哪些动物；2.查看某种动物的介绍信息；3.购票；4.查看路线推荐；5.查看最近园区；6.查看到某园区的距离
依靠引用Tourism_Manage的对象实现一些功能
Tourism_Manage
- old_ticket_sold : int = 0
- kid_ticket_sold : int = 0
- adult_ticket_sold : int = 0
- daily_income : long long = 0
- total_sold : long long = 0
- old_price : int
- kid_price : int
- adult_price : int
- family_price : int
+ Tourism_Manage(int old,int kid,int adult,int family) : constructor
+ show_daily_income() : void
+ show_old_sold() : void
+ show_kid_sold() : void
+ show_adult_sold() : void
+ show_family_sold() : void
+ show_total_num() : void
+ serve_order() : void
serve_order函数用来满足客户订单，让用户输入参观人员身份和相应类别人群的人数（老人小孩成人），并自动推荐最省钱选项，完成订单交易后，对订单数据记录
Land_Area
- contain : vector<ANIMAL>
- plant : string
- climate : string
- temp : float
- humidity : float
+ Land(const string&name, const string&plt, const string& clmt, float t, float hmd) : constructor
+ ~Land() : destructor
+ add_animal( ) : void
+ rmv_animal( ) : void
+ health_check( ) : void
+ feed_check( ) : void
+ show_park_species( ) : void
+ show_keeper( ) : void
+ show_animal_info( ) : void
+ show_environment( ) : void
Aqua_Area
- contain : vector<ANIMAL>
- water : string
- water_temp : float
+ Aqua(const string& name, const string& wt, float temp) : constructor
+ ~Aqua() : destructor
+ add_animal( ) : void
+ rmv_animal( ) : void
+ health_check( ) : void
+ feed_check( ) : void
+ show_park_species( ) : void
+ show_keeper( ) : void
+ show_animal_info( ) : void
+ show_environment( ) : void
Tourist
- tourism_manage : Tourism_Manage
+ Tourist() : constructor
+ ticket_service(const Tourism_Manage& tourism)
+ search_animal(const string&which_type)
+ search_park(const string&where)
+ recommend_route( )
+ show_nearest( )
+ show_distance( )
Keeper
- keeper_staff : vector<Worker_Info>
+ login() : bool
+ health_repo() : void
+ feed_repo() : void
Admin
- admin_staff : vector<Staff>
- tourism_manage : Tourism_Manage
+ login() : bool
+ add_keeper(const string& name,int id,int passwd,const vector<Staff>& keeper) : void
- rmv_keeper(int id, const vector<Staff>& keeper) : void
+ add_admin(const string& name,int id,int passwd) : void
- rmv_admin(int id) : void
+ check_ticket();