

Class 程序CRUD

String 程序保存路径 = 固定地址 + 词条AID + 词条BID;

String 程序文件名 = 程序数据ID + ".program";

目的：标准化处理

String 追加程序(Map<key, vaule> 程序信息){

// 取得程序的词条ID

程序词条ID = 词条.取得词条ID_by词条名("程序");

建立关系

// 取得Map.value的Type(Object Map.value)

// 如果Map.key不是"值" 且 如果Map.value是List时, 且List中是Map时

// 程序数据ID = 词条.采番数据ID_by词条ID(程序词条ID);

// 设置G词条信息(程序词条ID, 程序数据ID);

// 追加GW关系信息(List<Map>, G词条信息)

// 如果Map.key是"值" 且 Map.value是List时, 且List中是String时

// 程序数据ID = 词条.采番数据ID_by词条ID(程序词条ID);

// 需要以文件形式保存

// 需要有文件名、路径

// 真正的值是文件的全路径啊

// String 全路径 = 做成全路径(文件名, 路径)

// 追加文件(String 全路径, (List)Map.value)

除了程序之词条外。未设置List类型的Value

// 如果Map.key是"值" 且 如果Map.value是String时

// G词条信息 = 设置词条信息

// 程序数据ID = 追加实体数据(G词条信息, Map.value), 且返回数据采番ID

// return 程序数据ID

}

String 追加GW关系信息(List 程序信息){

// 分别取得Map.value.List的每个值, 进行处理 {

// W词条名 = 取得子词条名 (词条A(W), 词条B(W), 说明(W), 文件(W))

// W词条数据ID = (递归调用)追加程序(Map<key, vaule> 子程序信息)

// G词条信息 = 设置词条信息(词条ID=, 数据ID=程序 (数据ID))

// W词条信息 = 设置词条信息(词条ID=, 数据ID=词条A (词条ID))

// 追加Waiter关系(G词条信息, W词条信息);

// 追加Guest关系(G词条信息, W词条信息);

}

void 追加Waiter关系(G词条信息(词条ID=, 数据ID=程序 (数据ID))

void 追加Guest关系(G词条信息(词条ID=, 数据ID=程序 (数据ID))

String 取得Map.value的Type(Object Map.value)

Map 设置词条信息(词条ID=, 数据ID=程序 (数据ID))

String 追加实体数据(G词条信息, Map.value)

void 追加文件(String 全路径, (List)Map.value)

String 做成全路径(文件名, 路径)

return 文件名 + "¥" + 路径;

// 程序(G)----词条A(W)

// 程序(G)----词条B(W)

// 程序(G)----说明(W)

// 程序(G)----文件(W)