

Объект "Mailboxes" (версия на python 3.x)

Объект создан для офлайн-тестирования межагентного взаимодействия (МAB) роботов в среде программирования TrikStudio (на контроллерах Трик), но при необходимости его использование возможно и в других системах.

Основная идея заключается в том, чтобы была возможность тестировать МAB офлайн, а при запуске программы на реальных роботах менять только название основного объекта **mailboxes** -> **mailbox**, без изменения всего кода программы (или минимум). В связи с чем, названия основных методов MAILBOXES аналогичны методам в mailbox:

`.send()`, `.receive()`, `.hasMessages()`, `.myNullNumber()`

Каждый робот запускается в отдельной программе TrikStudio.

Объект **Mailboxes** использует модули `json` и `os`. До обращения к объекту необходимо объявить переменные:

`robotNulls = [1,10,20]` - массив с "бортовыми номерами" роботов, участвующих во взаимодействии
`robotNumber = 1` - "бортовой номер" текущего робота (должен меняться в программе у каждого робота)

Описание примера работы МAB (*mailboxes_example.py* или *mailboxes_aio.py*):

1. При необходимости меняются пути:
 - a. К расположению «файлообменника», по умолчанию путь "c:\temp\mailboxes"
 - b. При необходимости меняется путь к файлу "mailboxes_offline.py"
 - c. **Все пути указываются через прямой слэш '/' или двойной обратный слэш '\\'**
2. Запускаются отдельные копии TrikStudio (TS) в количестве = количеству роботов.
3. В каждой копии TS открывается скрипт 'mailboxes_example.py' или 'mailboxes_aio.py', где меняется **ТОЛЬКО** номер текущего робота - переменная `robotNumber`.
4. На дисплее симулятора контроллера Трик выводится надпись с номер каждого робота.
5. На симуляторе контроллера Трик, у робота с бортовым номер '1', нажимаем кнопку «стрелка вверх»
6. После чего «Робот 1» отправляет сообщение «Роботу 2»
7. «Робот 2» после получения сообщения проезжает вперед 2 клетки и отправляет сообщение «Роботу 3»
8. «Робот 3» после получения сообщения проезжает вперед 2 клетки и отправляет сообщение «Роботу 1»
9. «Робот 1» после получения сообщения проезжает вперед 2 клетки и отправляет сообщение всем роботам
10. Все роботы после получения сообщения одновременно едут вперед 2 клетки

Happy end ☺

Необходимые файлы:

mailboxes_offline.py	объект для офлайн – "общения" роботов, ссылка
mailboxes_example.py	основная программа для демонстрации межагентного взаимодействия, ссылка
mailboxes_aio.py	объект mailboxes и демокод в одной программе, ссылка
mav_python.mov	видео примера работы скриптов выше, ссылка

Свойства объекта *mailboxes*:

Наименование	Тип	Содержание
storagePath	Строка	Локальный / сетевой путь до размещения рабочих файлов
fileExt	Строка	Расширение имен файлов для обмена сообщениями
myFile	Строка	Полное имя файла для текущего робота (путь + название + расширение)

Методы (функции) объекта *mailboxes*:

Наименование	Параметры		Содержание
	Входные	Выходные	
<code>__init__</code>	<code>robotNumber</code> <code>[robotHulls]</code>	-	<p>1. Формирует свойства объекта:</p> <ul style="list-style-type: none"> <code>storagePath</code> - полный путь до расположения файлов <code>fileExt = '.txt'</code> - расширение файлов <code>robotNumber</code> - «бортовой» номер текущего робота <code>robotHulls</code> - список всех номеров роботов в команде <code>myFile</code> - имя файла для текущего робота <p>2. Удаляет файлы, соответствующие номерам роботов (номера берутся из свойства <code>robotHulls</code>)</p>
<code>myHullNumber</code>	-	<i>number</i>	Возвращает «бортовой номер» текущего робота (по сути значение переменной <code>myHullNumber</code> . Сделано для совместимости с основным объектом <code><mailbox></code>).
<code>hasMessages</code>	-	<i>boolean</i>	Проверяет наличие сообщений для текущего робота и возвращает <code><true></code> или <code><false></code>
<code>receive</code>	-	<i>array</i>	<p>Получает сообщения для текущего робота. Возвращает одномерный массив, где каждый элемент – строка из файла. Если сообщений нет, то будет ждать до тех пор, пока не появится хотя бы одно сообщение.</p> <p>Для обработки сообщений их необходимо «развернуть» из формата JSON командой <code>JSON.loads()</code></p>
<code>send</code>	<code>robots</code> (<i>list/number</i>), <code>msg</code> (<i>string/</i> <i>list</i>)	-	<p>Формирует и отправляет указанным роботам сообщения. Если не указаны номера роботов, сообщения отправляется всем (из массива номеров <code>robotHulls</code>), примеры:</p> <ul style="list-style-type: none"> ➤ <code>[]</code> – сообщение отправится всем роботам ➤ <code>2</code> или <code>[2]</code> – сообщение отправится только роботу с бортовым номером 2 ➤ <code>[2, 3]</code> – сообщение отправится роботам с номерами 2 и 3 <p>Сообщение отправляется в формате JSON В конце сообщения добавляется перевод строки <code>'\n'</code></p>
<code>writeFile</code>	<code>file</code> (<i>string</i>), <code>msg</code> (<i>string</i>)	-	Записывает сообщение <code><msg></code> в файл <code><file></code> .
<code>clearFile</code>	<code>file</code> (<i>string</i>)	-	Создает новый пустой файл <code><file></code> .
<code>readFile</code>	<code>file</code> (<i>string</i>)	<i>data</i> (<i>array</i>)	Считывает все данные из файла <code><file></code> . Обнуляется файл <code><file></code> . Возвращает массив с данными, где каждый элемент - строка из файла.