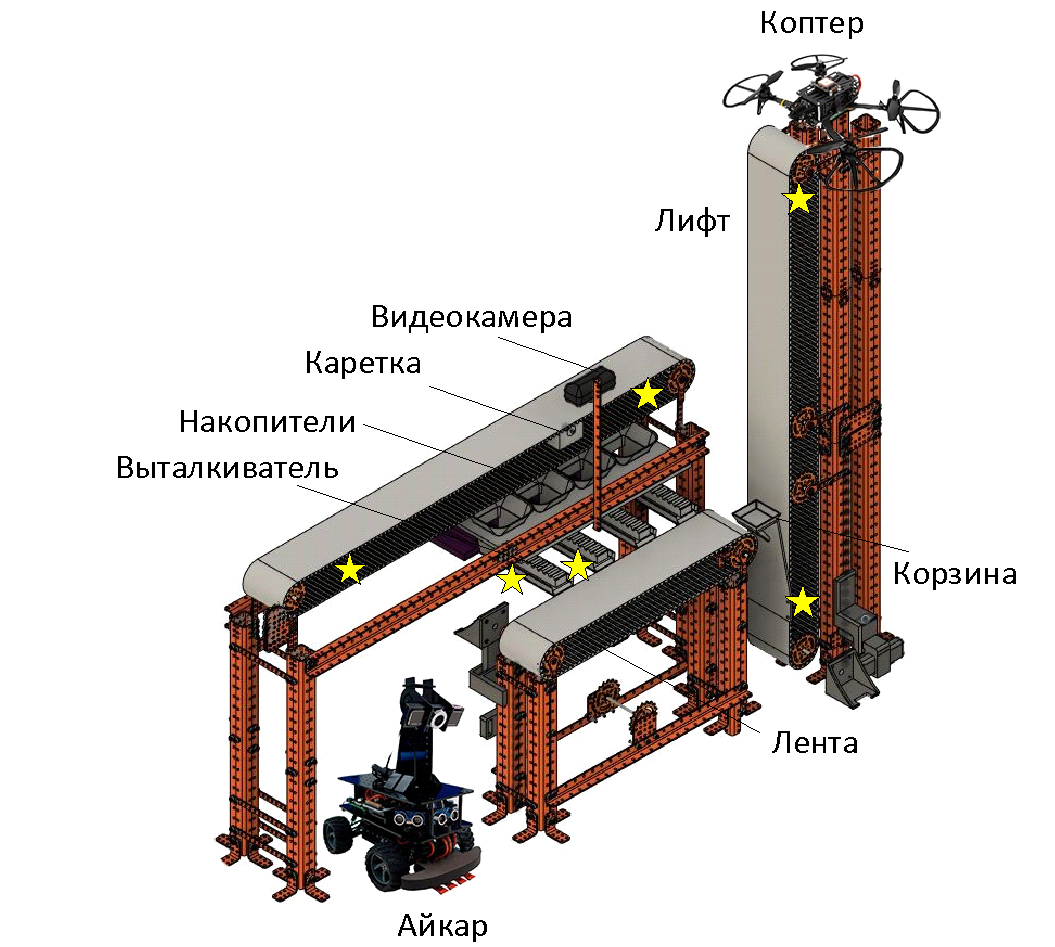
**Документация**

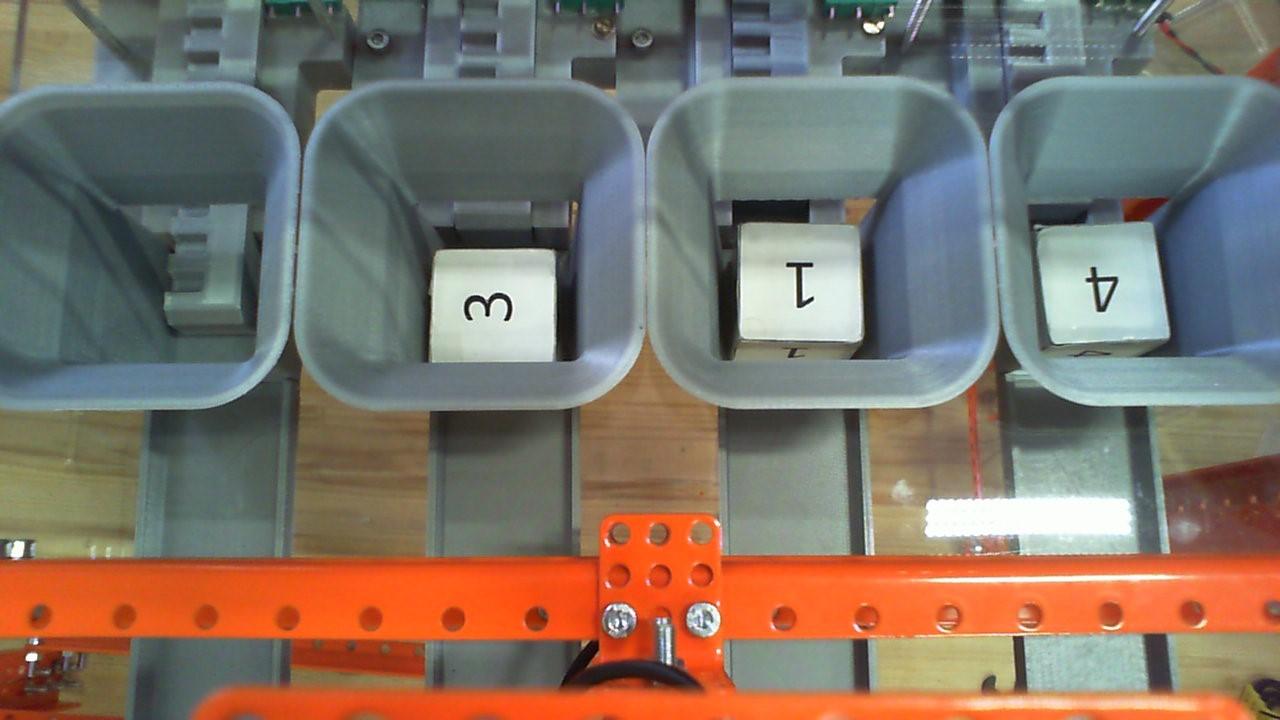
Для реализации сортировки кубиков в распределительном хабе участникам будет предоставлен набор функций, чей интерфейс специально сведен до вида функций, не принимающих никакие аргументы, с целью предоставить наиболее высокий («блочный») уровень работы с логикой установки. От участников не требуется понимания отдельных аспектов программной работы и «внутренней» логики обращения к отдельным узлам установки и общения с ними. В контексте текущего задания участникам необходимо разработать алгоритм сортировки, используя заготовленные нами «блоки» - набор уникальных функций для каждой из набора условно-элементарных задач.



***Рисунок 1****. Виртуальная модель фабрики (производственной ячейки) и ее узлов*

Установка обладает 4 узлами: **каретка с магнитом, лента конвейера, накопители и лифт**. Каждый узел управляется своими функциями. Процесс сортировки происходит следующим образом: условный «груз» представляет собой кубик, который доставляется при помощи Айкара непосредственно в зону выгрузки. Каретка с магнитом передвигается по горизонтальной оси, параллельной поверхности, на которой находится фабрика, и осуществляет спуск ленты, на конце которой находится магнит для захвата груза в точке его получения. После того как на магнит будет подано напряжение, действие магнитных сил вступит в силу, и груз будет успешно захвачен и готов для дальнейшего перемещения в пространстве. Далее магнит с уже захваченным грузом будет поднят на высоту, отвечающую следующему требованию: кубик не должен задевать в процессе дальнейшего горизонтального (параллельного поверхности, на которой находится производственная установка) передвижения конвейерную ленту или накопители. Как только данное условие будет удовлетворено, последующий алгоритм должен состоять из постепенного передвижения каретки с магнитом и захваченным им грузом до точки в пространстве, находящейся внутри зоны захвата видеокамеры, установленной в целях распознавания и последующей классификации объектов. Далее описанный участниками алгоритм распознавания грузов должен корректно задетектировать и соотнести груз с одной из групп (соответствуют номеру, расположенному на гранях этого самого груза) и, как следствие, соответствующим накопителем для груза.

С этого момента происходит условное разветвление в алгоритме: в зависимости от номера группы (от 0 до 4 включительно) груз, схваченный магнитом, передвигается вместе с кареткой в точку выгрузки. Как только магнит находится на одной вертикальной оси с соответствующим его номеру накопителем, должна быть произведена выгрузка: лента с магнитом должна быть опущена до того момента, пока груз не будет в непосредственной близости ко дну накопителя. После этого напряжение на магните должно быть отключено, что приведет к окончанию действия магнитных сил, и, как следствие, к выгрузке в накопитель. Далее лента с магнитом принимаем верхнее положение, откатывается обратно в начало и все повторяется с самого начала до тех пор, пока с Айкара не будут выгружены все кубики. Как только последний груз будет доставлен в соответствующий накопитель, выгрузка будет считаться законченной, а с ней, соответственно, и процесс сортировки груза.



***Рисунок 2****. Пример расположения кубиков в накопителях производственной установки в процессе сортировки*

Далее при помощи перемещения каретки с магнитом, активации магнита и переноса грузов на первый и единственный выталкиватель, который при помощи применения механического силового воздействия в порядке очереди по возрастанию числового значения группы грузов, находящихся в соответствующих накопителях, выталкивает груз на конвейерную ленту. На ней груз едет ровно до тех пор, пока не окажутся в корзине, установленной на вертикальной лифтовой ленте. Как только груз окажется в корзинке, лифт должен поднять груз на крышу для дальнейшего перемещения в пространстве при помощи квадрокоптера. Задача считается выполненной, когда а) грузы всех групп будут доставлены на крышу и переданы квадрокоптеру для дальнейшей транспортировки б) программа, написанная для распределительного хаба, будет выводить в командную строку последовательность цифр на грузах слева направо.

Участникам доступен набор из 8 уникальных функций для прямого обращения к отдельным узлам производственной установки. Существует возможность использовать любые средства языка программирования Python для написания универсальных функций с применением компьютерного зрения по распознаванию числовых обозначений на гранях кубиков. Кубики являются условным грузом, принятым с Айкара при помощи магнита и доставленным в соответствующие числовым обозначениям на гранях того или иного кубика накопители. Участники вольны использовать любой известный ему программный подход для отслеживания и классификации кубиков, полученных с Айкара и принятых при помощи магнита (например, написание нейронной сети).

Далее приведено подробное описание функций управления процессом сортировки с примерами их использования в реальном коде:>>

**Описание функций из facility\_api.py**

**1.** MagnetUp() **–** поднять магнит в верхнее положение, удовлетворяющее условию (кубик не должен задевать в процессе дальнейшего горизонтального передвижения конвейерную ленту или накопители). В окне отладки выводит название функции, если функция успешно завершилась;

**2.** MagnetHalfDown() **–** опустить магнит в специальное рабочее положение - на уровень накопителя для захвата груза из соответствующего накопителя или выгрузки очередного груза в соответствующий накопитель. В окне отладки выводит название функции, если функция успешно завершилась;

**3.** MagnetOn() **–** включить магнит при помощи подачи напряжения на него; во включенном состоянии магнит способен захватывать груз для дальнейшего перемещения в пространстве. Функция MagnetOff служит для обратного действия, отключая магнит

4.MagnetOff() **–** выключить магнит при помощи остановки подачи напряжения на него; в выключенном состоянии магнит не способен захватывать груз для дальнейшего перемещения в пространстве; иными словами, магнит находится в «спящем режиме», пока не будет вызвана функция MagnetOn;

5. ToFirstHoarder() - специальная функция перемещения каретки с магнитом в точку, лежащую на одной вертикальной оси с **первым** накопителем, обладающим выталкивателем для отправления грузов на конвейерную ленту. В окне отладки выводит название функции, если функция успешно завершилась;

6. ToSecondHoarder() - специальная функция перемещения каретки с магнитом в точку, лежащую на одной вертикальной оси со **вторым** накопителем, не обладающим выталкивателем для отправления грузов на конвейерную ленту. В окне отладки выводит название функции, если функция успешно завершилась;

7. ToThirdHoarder() - специальная функция перемещения каретки с магнитом в точку, лежащую на одной вертикальной оси с **третьим** накопителем, не обладающим выталкивателем для отправления грузов на конвейерную ленту. В окне отладки выводит название функции, если функция успешно завершилась;

8. ToFourthHoarder() - специальная функция перемещения каретки с магнитом в точку, лежащую на одной вертикальной оси со **четвертым** накопителем, не обладающим выталкивателем для отправления грузов на конвейерную ленту. В окне отладки выводит название функции, если функция успешно завершилась;

**\*Дополнительное описание функций из facility.py**

**1.** MagnetUp() **–** поднять магнит в верхнее положение, удовлетворяющее условию (кубик не должен задевать в процессе дальнейшего горизонтального передвижения конвейерную ленту или накопители). В окне отладки выводит название функции, если функция успешно завершилась;

**2.** MagnetHalfDown() **–** опустить магнит в специальное рабочее положение - на уровень накопителя для захвата груза из соответствующего накопителя или выгрузки очередного груза в соответствующий накопитель. В окне отладки выводит название функции, если функция успешно завершилась;

**3.** MagnetFullDown() **–** опустить магнит на уровень грузового отсека Айкара для захвата груза из грузового автономного автомобиля. В окне отладки выводит название функции, если функция успешно завершилась;

**4.** MoveTo() **–** общая функция управления шаговыми двигателями каретки с магнитом для ее передвижения по горизонтальной оси, параллельной основе (упору) производственной фабрики на определенное количество шагов, определяющее конечную точку в траектории перемещения каретки. Используется в более конкретных функциях по передвижению: ToFirstHoarder, ToSecondHoarder, ToThirdHoarder, ToFourthHoarder;

**5.** MagnetOn() **–** включить магнит при помощи подачи напряжения на него; во включенном состоянии магнит способен захватывать груз для дальнейшего перемещения в пространстве. Функция MagnetOff служит для обратного действия, отключая действия магнита;

6.MagnetOff() **–** выключить магнит при помощи остановки подачи напряжения на него; в выключенном состоянии магнит не способен захватывать груз для дальнейшего перемещения в пространстве; иными словами, магнит находится в «спящем режиме», пока не будет вызвана функция MagnetOn;

7. ToFirstHoarder() - специальная функция перемещения каретки с магнитом в точку, лежащую на одной вертикальной оси с **первым** накопителем, обладающим выталкивателем для отправления грузов на конвейерную ленту. В окне отладки выводит название функции, если функция успешно завершилась;

8. ToSecondHoarder() - специальная функция перемещения каретки с магнитом в точку, лежащую на одной вертикальной оси со **вторым** накопителем, не обладающим выталкивателем для отправления грузов на конвейерную ленту. В окне отладки выводит название функции, если функция успешно завершилась;

9. ToThirdHoarder() - специальная функция перемещения каретки с магнитом в точку, лежащую на одной вертикальной оси с **третьим** накопителем, не обладающим выталкивателем для отправления грузов на конвейерную ленту. В окне отладки выводит название функции, если функция успешно завершилась;

10. ToFourthHoarder() - специальная функция перемещения каретки с магнитом в точку, лежащую на одной вертикальной оси со **четвертым** накопителем, не обладающим выталкивателем для отправления грузов на конвейерную ленту. В окне отладки выводит название функции, если функция успешно завершилась;

11. LiftUp() - специальная функция перемещения лифта с корзиной для перемещения на площадку для посадки квадрокоптера и дальнейшей передачи этого груза квадрокоптеру, находящуюся на крыше башни. В окне отладки выводит название функции, если функция успешно завершилась;

12. LiftDown() - специальная функция перемещения лифта с корзиной для передачи груза на нижний уровень для получения нового поступающего груза в пространство корзины;

**13.** CarriageStart() - функция перемещения каретки в стартовую зону для дальнейшего получения груза, прибывшего в зону выгрузки на Айкаре. В окне отладки выводит название функции, если функция успешно завершилась;

**14.** CarriageEnd() - функция перемещения каретки в конечную зону. В окне отладки выводит название функции, если функция успешно завершилась;

**15.** LentaUp() - функция перемещения конвейерной ленты;

**16.** Servo() - функция перемещения выталкивателя для отправки грузов из первого накопителя на конвейерную ленту. В зависимости от типа производимых действий (втягивание и выталкивание) в качестве отладки могут прийти следующие ключевые слова: ’ServoBackward’ и ’ServoForward’.

\**Примечание:* ***участникам олимпиады строго запрещается внесение любых изменений в файл facility.py, а также использование описанных там сервисных функций!***