Сканирование собранных заказов для отгрузки в Озон.

Выполнил студент 136 группы

Горецкий Денис Владимирович

2024.

Содержание

1.   
 1.1 Обоснование выбора темы……………………………………….………. 3  
 1.2 Определение цели и задач исследования...……………………………… 3

2. Ключевые компоненты ...………… ………………………………………… 4

3. Архитектура кода ...………………… ……………………………………… 4

4. Работа кода...……………………… ………………………………………… 5

5. Заключение...…………………… …………………………………………… 9

6. Список необходимых программ установленных на компьютер. ………… 10

7. Код ...………………………………… ……………………………………… 10

**1.1** **Обоснование выбора темы.**

В связи с возросшим количеством заказов на отгрузку в Озон (больше 50-100 в день), на данный момент до 500 в день и не возможностью быстро подключить и настроить предлагаемое сейчас на рынке программное обеспечение с имеющейся у фирмы программой 1С (НЕ управляемые формы), был написан код работающий в Pycharm. Для того что бы позволить складу работать в штатном режиме, так как просчитать такое количество заказов в ручном режиме оказалось практически невозможно, очень много ошибок, а так же, найти отправление с неправильным (плохо распечатался, заказ не из отгрузки на сегодня, заказ от которого отказался клиент и т.д.)

* 1. **Определение целей и задач:**

Позволить кладовщику:

Отсканировать сканером штрих-код отгружаемого заказа, после чего, по звуковому сопровождению и выводу в консоль кладовщик понимает:

1. Заказ должен попасть в данную отгрузку! – Все Ок.
2. Или заказ попал в собранные «случайно», и должен был уехать ранее или должен уехать в следующий раз, или клиент отказался от заказа и он должен быть разобран и товар при этом возвращается назад на полку, или не читается штрих-код (сбой принтера, плохое качество бумаги, красящей ленты)
3. После того как все имеющиеся заказы отсканированы, кладовщик видит совпало количество с требуемым для отгрузки или нет.
4. При каждом сканирование после звукового сигнала (звуковые сигналы разные) на экран выводиться количество отсканированных и оставшихся не отсканированными заказов.
5. Если какого-то количества заказов не хватает, а доступные для сканирования заказы закончились, кладовщик вводит 4 и на экран выводиться информация по всем недостающим заказам, где видно Номер заказа в 1С, номер штрих-кода и номер отправления.
6. Далее кладовщик разбирается почему заказ(ы) не находятся в месте отгрузки и либо забирает их у сборщиков либо распечатывает в 1С листы сборки для дальнейшего комплектования заказов.
7. Так же данная процедура исключила такую проблему как отказ от доставки в связи с тем что на распределительном центре Озона не читается шрих-код.
8. После внедрения данной процедуры, прекратились проблемы недоставки заказов от нас до распределительно центра Озон.
9. **Ключевые компоненты.**

Библиотека winsound, позволяет воспроизводить звуковые сигналы для различных ситуаций.

Так же код включает следующие компоненты, несколько переменных, списков, циклов for и while словарь и функцию:

1. **Архитектура кода.**

Архитектура приложения:

1. list\_not\_find = [] – Список,

в который добавляются не найденные заказы (это для дальнейшей доработки программы, после которой заказы будут отправляться на печать листов сборки в 1С, через базу данных или как то иначе), так же эта информация будет полезна для анализа работы склада в будущем.

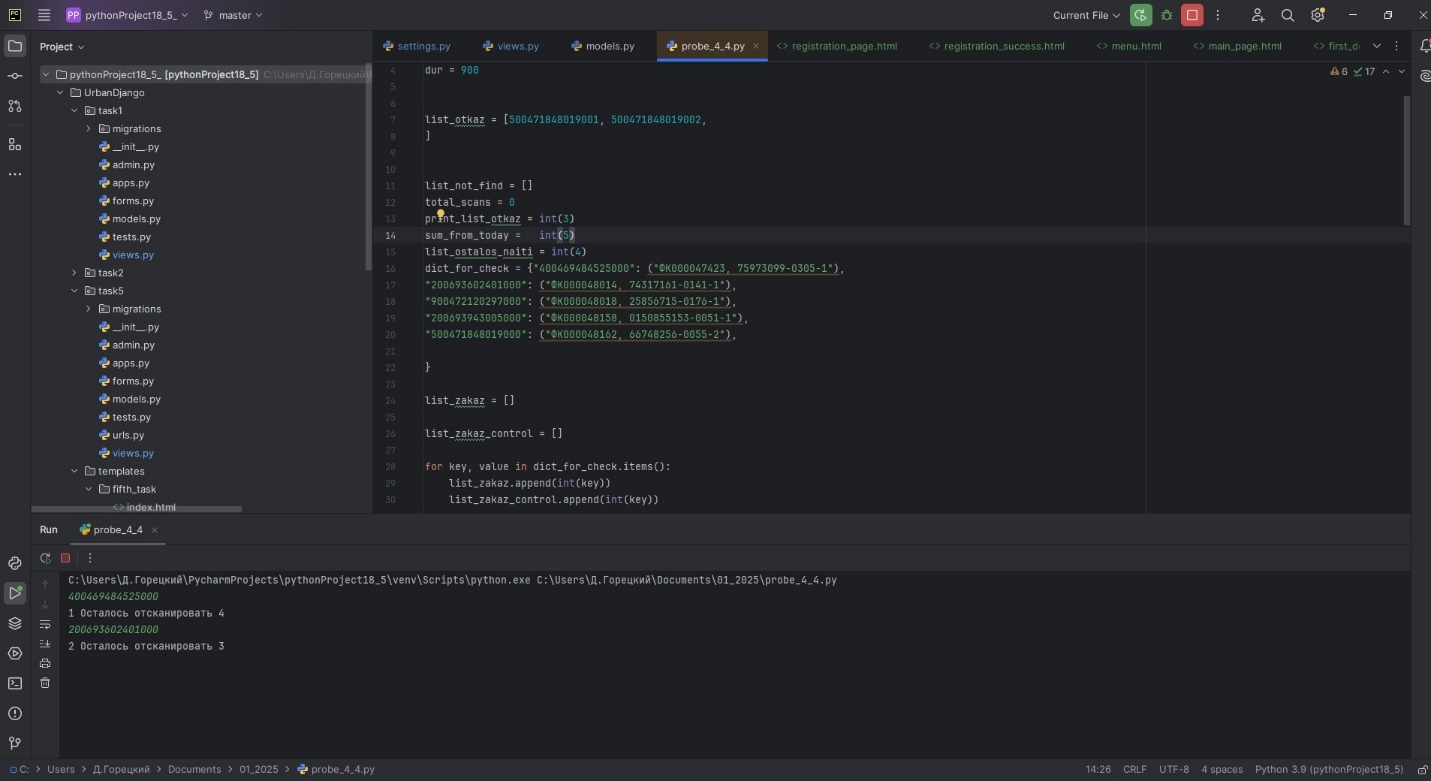
1. total\_scans = 0 - int -, кол-во отсканированных заказов.
2. print\_list\_otkaz = int(3) - Переменная - при вводе кладовщиком значения которой выводиться на экран заказы которые нужно убрать из доставки. Сейчас это оказалось не так актуально, но в будущем может быть полезным, для анализа работы склада.
3. sum\_from\_today = int(5) – Сумма всех отсканированных сегодня заказов (включая «отказы» и «лишние» заказы), так же полезно, для анализа работы склада.
4. list\_ostalos\_naiti = int(4) – ввод значения этой переменной выводит на экран информацию о тех заказах, которые еще небыли отсканированы.
5. dict\_for\_check = { } – словарь куда вставляются:
6. { "800586752593000": ("ФК000067142, 86030681-0163-2"),} из подготовленного файла вставляются ключ (штрих-код в числовом отображении) и значение состоящие из номера заказа в 1С и номер отправления.
7. list\_otkaz = [] список с номерами штрихкодов «отказных» заказов
8. list\_zakaz = [] – список создающийся при запуске модуля в котором храниться номера штрих-кодов, из которого при сканировании удаляются номера отсканированных штрих-кодов.
9. list\_zakaz\_control = [] – контрольный список создающийся при запуске модуля в котором храниться номера штрих-кодов. Так же в будущем это будет полезно для анализа работы склада.
10. Цикл for для создания списков list\_zakaz = [] и list\_zakaz\_control = []
11. Основная функция обрабатывающая ввод со сканера и так же ввод информации с клавиатуры. - def pic(sh\_cod):
12. total\_scans = 0, при сканировании «правильного» штрих-кода, прибавляется +1, для дальнейшего вывода на консоль. Добалена в функцию def pic(sh\_cod): global total\_scans как глобальная переменная.
13. Цикл while: который перезапускает функцию def pic(sh\_cod): - до тех пор пока кладовщик не выйдет из модуля, в который добавлен блок обработки исключений (exception handling) призванный «отловить» не корректный ввод (при сканировании не корректного шрих-кода)
14. **Работа кода.**

Перед запуском модуля, менеджер маркетплейсов по шаблону создает exel файл с информацией о сегодняшней отгрузке и выкладывает его в общий доступ.

Кладовщик запускает pycharm, и в модуле с кодом, добавляет скопированный из файла exel столбец и вставляет в словарь dict\_for\_check = { }, если есть номера отказов – он их копирует и вставляет в список list\_otkaz = []

После чего запускает модуль и начинает сканирование собранных заказов.

При сканировании «правильного» заказа программа воспроизводит определенный звук и выводит на экран информацию о том, сколько заказов отсканировано и сколько осталось отсканировать, так же удаляя введённый (отсканированный) номер из списка list\_zakaz созданного ранее:

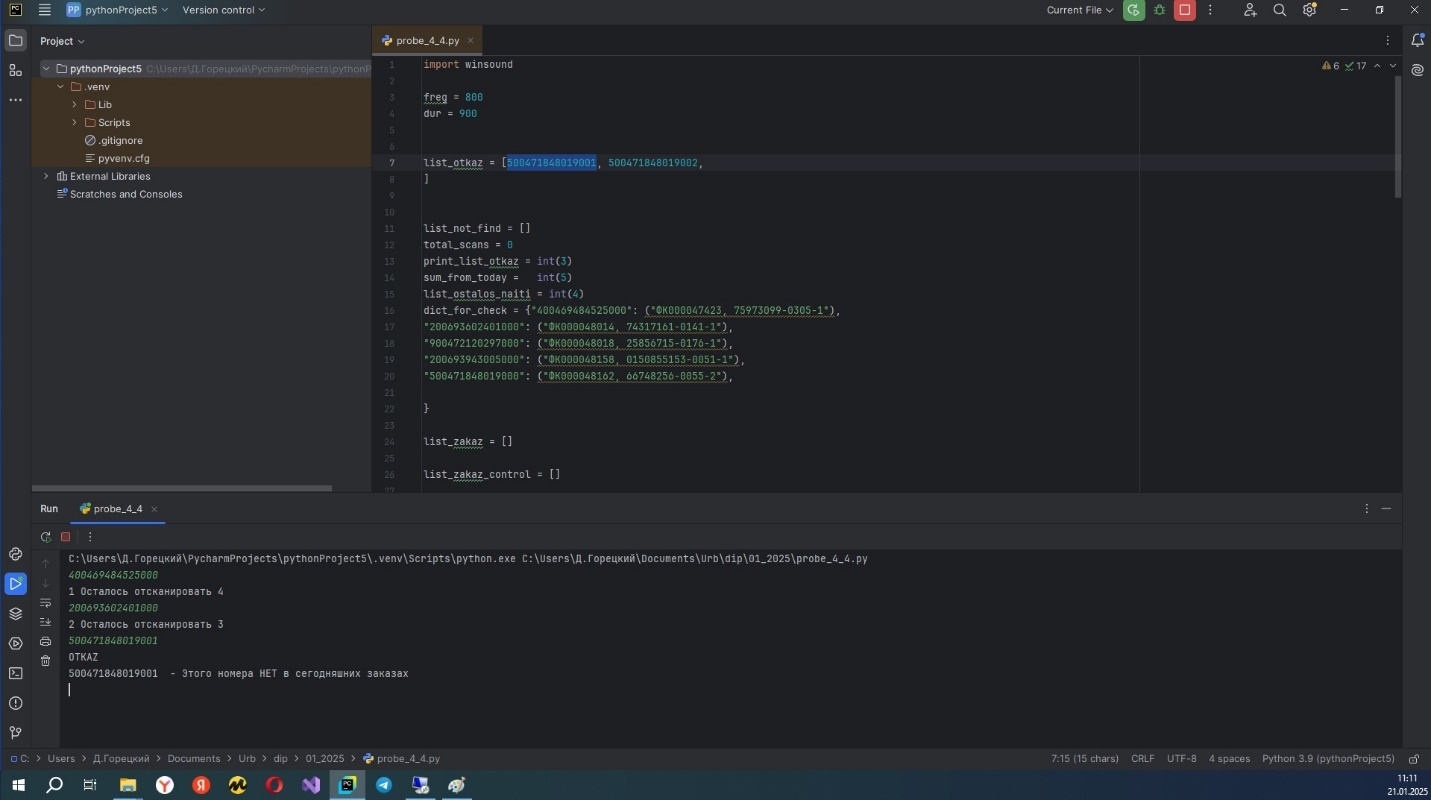


Если был отсканирован штрих-код с заказом от которого отказались (номер которого находиться в списке list\_otkaz = [] - на консоль выводиться строка

OTKAZ

и

номер шрих-кода - Этого номера НЕТ в сегодняшних заказах и воспроизводиться звук говорящий о том, что то пошло не так.



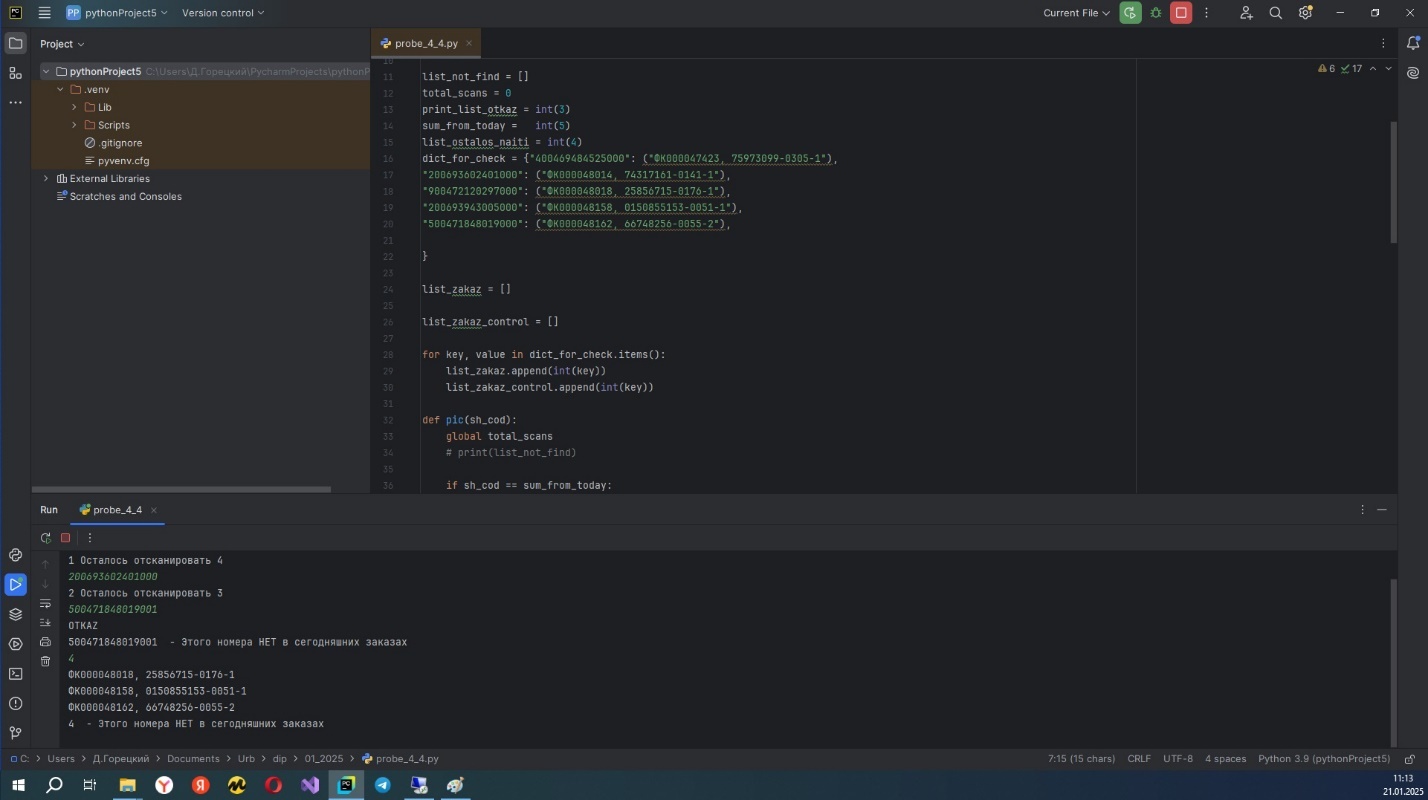
Если все доступные заказа отсканированы а в консоли кладовщик видит, что еще осталось отсканировать какое то количество заказов, он на клавиатуре вводит 4, что вызывает обращение к части кода с переменной list\_ostalos\_naiti = int(4)

‘’’

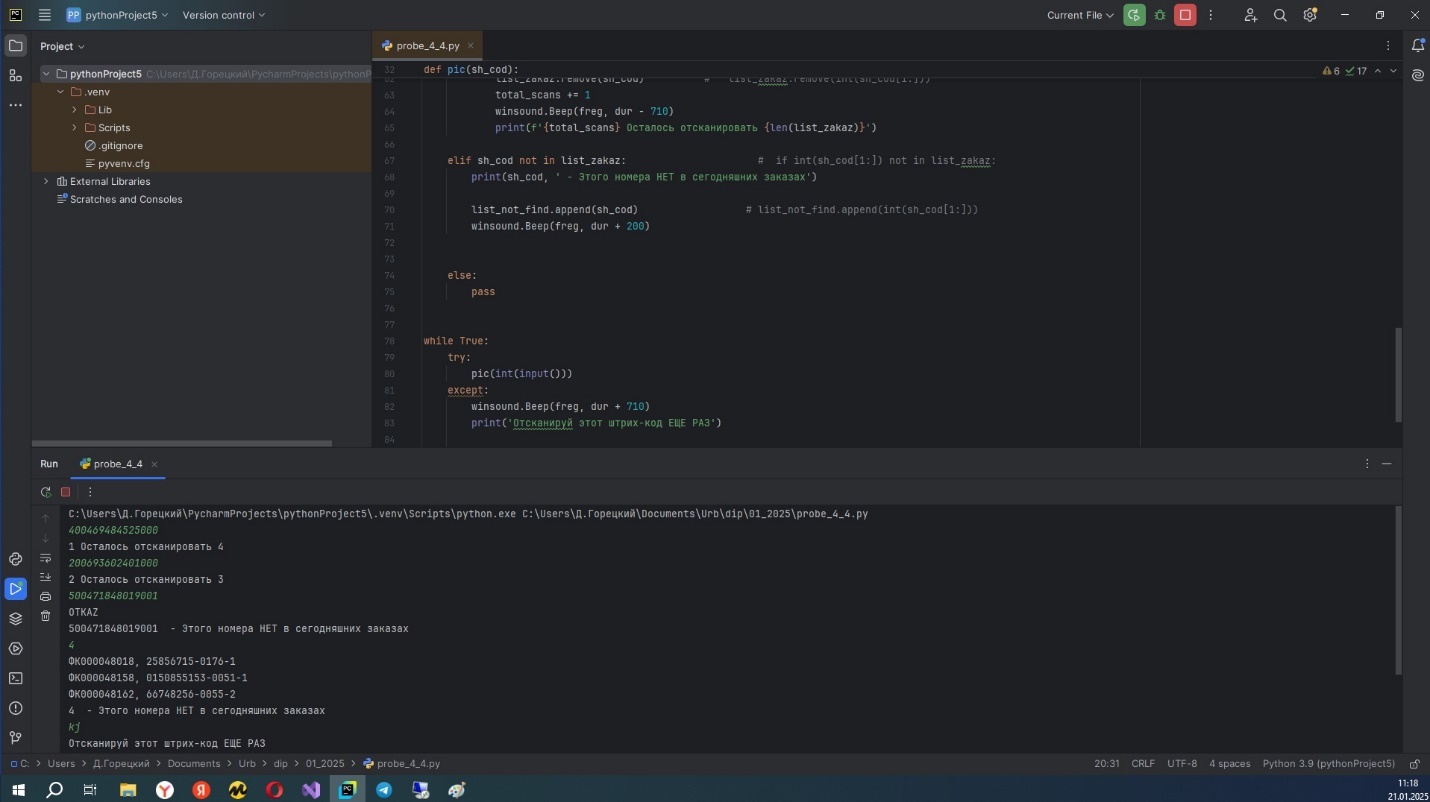
if sh\_cod == list\_ostalos\_naiti:  
 for i in list\_zakaz:  
 print(dict\_for\_check[str(i)])

‘’’

После чего цикл for берет поочередно все оставшиеся заказы из list\_zakaz и по этому номеру - ключу распечатывается информация из словаря dict\_for\_check = { }

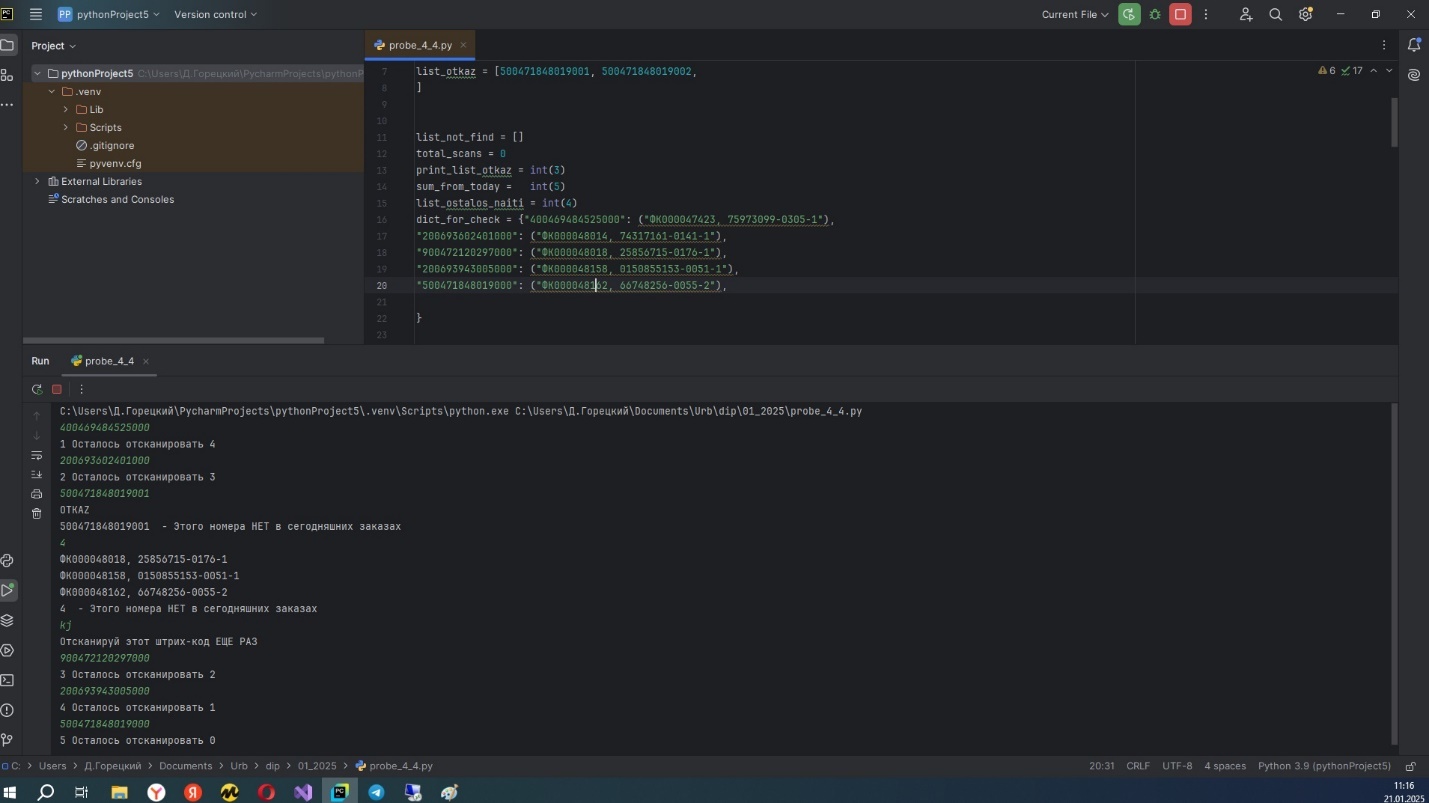


Если был отсканирован не корректный номер штрих-кода (плохо распознал сканер, был плохо распечатан и т.д.) звучит определенный звуковой сигнал и на консоль выводиться информация – которая была распознана сканером и «Отсканируй этот штрих-код ЕЩЕ РАЗ»



После того как все заказы отсканированы, и на консоли выведена информация

…. Осталось отсканировать 0,



кладовщик нажимает Ctrl + F2 и удаляет информацию из list\_otkaz = [] и dict\_for\_check = { }

1. **Заключение**

Этот код был написан для того, что бы позволить складу работать в штатном режиме, исключить возврат заказов с плохо читаемым штрих-кодом, понимать что Все заказы были отгружены и небыли отгружены лишние, а так же выявлять те заказы, которые по какой то причине небыли собраны (не распечатался по какой то причине Лист Сборки или Расходный ордер, или кто то потерял эти документы или забыли положить заказ в доставку) и принимать меры по устранению этих проблем, выработке условий (правил, требований к персоналу склада) при которых такие проблемы исключаются (это в постоянном процессе).

1. **Список необходимых библиотек и программ установленных на компьютер**

Встроенная библиотека winsound

**Python 3.9**

**PyCharm 2024.2 (Community Edition)**

1. **Код:**

```

import winsound  
  
freg = 800  
dur = 900  
  
  
list\_otkaz = [500471848019001, 500471848019002,  
]  
  
list\_not\_find = []  
total\_scans = 0  
print\_list\_otkaz = int(3)  
sum\_from\_today = int(5)  
list\_ostalos\_naiti = int(4)  
dict\_for\_check = {"400469484525000": ("ФК000047423, 75973099-0305-1"),  
"200693602401000": ("ФК000048014, 74317161-0141-1"),  
"900472120297000": ("ФК000048018, 25856715-0176-1"),  
"200693943005000": ("ФК000048158, 0150855153-0051-1"),  
"500471848019000": ("ФК000048162, 66748256-0055-2"),  
  
}  
  
list\_zakaz = []  
  
list\_zakaz\_control = []  
  
for key, value in dict\_for\_check.items():  
 list\_zakaz.append(int(key))  
 list\_zakaz\_control.append(int(key))  
  
def pic(sh\_cod):  
 global total\_scans  
 # print(list\_not\_find)  
  
 if sh\_cod == sum\_from\_today:  
 print(f'Отсканировано "Сегодняшних" заказов: {total\_scans}, осталось отсканировать {len(list\_zakaz)}')  
  
 if sh\_cod in list\_not\_find: # if int(sh\_cod[1:]) in list\_not\_find:  
 print('Этот заказ уже БЫЛ Отсканирован сегодня')  
 winsound.Beep(freg, dur + 100)  
  
 if sh\_cod in list\_otkaz: # if int(sh\_cod[1:]) in list\_otkaz:  
 print('OTKAZ')  
 list\_otkaz.remove(sh\_cod)  
 # print(list\_zakaz, 'list\_zakaz - 2')  
 winsound.Beep(freg, dur + 500)  
  
 if sh\_cod not in list\_zakaz and sh\_cod in list\_zakaz\_control:  
 print('Этот номер уже БЫЛ Отсканирован сегодня')  
 winsound.Beep(freg, dur + 800)  
  
 if sh\_cod == list\_ostalos\_naiti:  
 for i in list\_zakaz:  
 print(dict\_for\_check[str(i)])  
  
  
 if sh\_cod == print\_list\_otkaz:  
 print(f'Осталось найти "Отказных" заказов {len(list\_otkaz)}, с номерами штрих-кодов: {list\_otkaz}')  
  
 elif sh\_cod in list\_zakaz:  
  
 list\_zakaz.remove(sh\_cod) # list\_zakaz.remove(int(sh\_cod[1:]))  
 total\_scans += 1  
 winsound.Beep(freg, dur - 710)  
 print(f'{total\_scans} Осталось отсканировать {len(list\_zakaz)}')  
   
 elif sh\_cod not in list\_zakaz: # if int(sh\_cod[1:]) not in list\_zakaz:  
 print(sh\_cod, ' - Этого номера НЕТ в сегодняшних заказах')  
  
 list\_not\_find.append(sh\_cod) # list\_not\_find.append(int(sh\_cod[1:]))  
 winsound.Beep(freg, dur + 200)  
   
   
 else:  
 pass  
  
  
while True:  
 try:  
 pic(int(input()))  
 except:  
 winsound.Beep(freg, dur + 710)  
 print('Отсканируй этот штрих-код ЕЩЕ РАЗ')

```