

Project

Hydronic Riser Sizing Program

Using Python and Object-Oriented Programming Techniques

Dildeep Dhillon
12-1-2022

Introduction:

Used Python and the Pandas library to extract information for an Excel workbook, then this information was stored and organized into vectors that contained objects (objects are created from the information we extracted). These objects are then updated with new values for the size of the piping, and flow coming in for each riser and each floor, which is calculated based on which level the supply flow comes in. The goal of the program is to take the information from the Excel workbook and a few inputs from the user to generate an Excel workbook that lists the pipe size, and flow (in gpm) for each floor and each riser. Which improves workflow for engineering sales organizations, automating tedious multi-hour tasks and eliminating human error.

Input Excel File (only portion of file shown):

FLOOR	RISER	SUITE	TAG		
10	HPS-1	1001	HP-2		
15	HPS-1	1501	HP-2		
19	HPS-1	1901	HP-2		
1	HPS-2	101	HP-2		
2	HPS-1	201	HP-2		
3	HPS-1	301	HP-2		
4	HPS-1	401	HP-2		
5	HPS-1	501	HP-2		
6	HPS-1	601	HP-2		
7	HPS-1	701	HP-2		
8	HPS-1	801	HP-2		
9	HPS-1	901	HP-2		
2	HPS-2	101	HP-2		
11	HPS-1	1101	HP-2		
4	HPS-2	401	HP-2		
6	HPS-2	601	HP-2		
5	HPS-2	501	HP-2		
12	HPS-1	1201	HP-2		
13	HPS-1	1301	HP-2		
14	HPS-1	1401	HP-2		
1	HPS-2	202	HP-2		
16	HPS-1	1601	HP-2		
17	HPS-1	1701	HP-2		
18	HPS-1	1801	HP-2		
3	HPS-2	301	HP-5		
20	HPS-1	2001	HP-2		
21	HPS-1	2101	HP-2		
22	HPS-1	2201	HP-2		
23	HPS-1	2301	HP-2		
24	HPS-1	2401	HP-2		
25	HPS-1	2501	HP-2		
26	HPS-1	2601	HP-2		
27	HPS-1	2701	HP-2		
28	HPS-1	2801	HP-2		
29	HPS-1	2901	HP-2		
30	HPS-1	3001	HP-2		

< >

TAKEOFF

SCHEDULE

+

	A	B	C	D
1	TAG	MODEL	FLOW	
2	HP-2	UVHP-ERV-04	2	
3	HP-5	UVHP-ERV-06	3	
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				
16				
17				
18				
19				
20				
21				
22				
23				
24				
25				
26				
27				
28				
29				
30				
31				
32				
33				
34				
35				
36				
37				

< >

TAKEOFF

SCHEDULE

Output Excel File (only portion of file shown):

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Floor	Risers	Suite	Tag	Model	Flow	Supply Flow Below	Supply Flow Top	Supply Size Bottom	Supply Size Top	supply %		
2	36	HPS-1	3601	HP-5	UVHP-ERV-06	3	3	0	3/4	N/A	3%		
3	35	HPS-1	3501	HP-5	UVHP-ERV-06	3	6	3	1	3/4	6%		
4	34	HPS-1	3401	HP-2	UVHP-ERV-04	2	8	6	1	1	8%		
5	33	HPS-1	3301	HP-2	UVHP-ERV-04	2	10	8	1 1/4	1	10%		
6	32	HPS-1	3201	HP-2	UVHP-ERV-04	2	12	10	1 1/4	1 1/4	12%		
7	31	HPS-1	3101	HP-2	UVHP-ERV-04	2	14	12	1 1/4	1 1/4	14%		
8	30	HPS-1	3001	HP-2	UVHP-ERV-04	2	16	14	1 1/4	1 1/4	16%		
9	29	HPS-1	2901	HP-2	UVHP-ERV-04	2	18	16	1 1/2	1 1/4	18%		
10	28	HPS-1	2801	HP-2	UVHP-ERV-04	2	20	18	1 1/2	1 1/2	20%		
11	27	HPS-1	2701	HP-2	UVHP-ERV-04	2	22	20	1 1/2	1 1/2	22%		
12	26	HPS-1	2601	HP-2	UVHP-ERV-04	2	24	22	1 1/2	1 1/2	24%		
13	25	HPS-1	2501	HP-2	UVHP-ERV-04	2	26	24	2	1 1/2	26%		
14	24	HPS-1	2401	HP-2	UVHP-ERV-04	2	28	26	2	2	28%		
15	23	HPS-1	2301	HP-2	UVHP-ERV-04	2	30	28	2	2	30%		
16	22	HPS-1	2201	HP-2	UVHP-ERV-04	2	32	30	2	2	32%		
17	21	HPS-1	2101	HP-2	UVHP-ERV-04	2	34	32	2	2	34%		
18	20	HPS-1	2001	HP-2	UVHP-ERV-04	2	36	34	2	2	36%		
19	19	HPS-1	1901	HP-2	UVHP-ERV-04	2	38	36	2	2	38%		
20	18	HPS-1	1801	HP-2	UVHP-ERV-04	2	40	38	2	2	40%		
21	17	HPS-1	1701	HP-2	UVHP-ERV-04	2	42	40	2	2	42%		
22	16	HPS-1	1601	HP-2	UVHP-ERV-04	2	44	42	2	2	44%		
23	15	HPS-1	1501	HP-2	UVHP-ERV-04	2	46	44	2	2	46%		
24	14	HPS-1	1401	HP-2	UVHP-ERV-04	2	48	46	2	2	48%		
25	13	HPS-1	1301	HP-2	UVHP-ERV-04	2	50	48	2 1/2	2	50%		
26	12	HPS-1	1201	HP-2	UVHP-ERV-04	2	52	50	2 1/2	2 1/2	52%		
27	11	HPS-1	1101	HP-2	UVHP-ERV-04	2	54	52	2 1/2	2 1/2	54%		
28	10	HPS-1	1001	HP-2	UVHP-ERV-04	2	56	54	2 1/2	2 1/2	56%		
29	9	HPS-1	901	HP-2	UVHP-ERV-04	2	58	56	2 1/2	2 1/2	58%		
30	8	HPS-1	801	HP-2	UVHP-ERV-04	2	60	58	2 1/2	2 1/2	60%		
31	7	HPS-1	701	HP-2	UVHP-ERV-04	2	62	60	2 1/2	2 1/2	62%		
32	6	HPS-1	601	HP-2	UVHP-ERV-04	2	64	62	2 1/2	2 1/2	64%		
33	5	HPS-1	501	HP-2	UVHP-ERV-04	2	66	64	2 1/2	2 1/2	66%		
34	4	HPS-1	401	HP-2	UVHP-ERV-04	2	34	36	2	2	34%		
35	3	HPS-1	301	HP-2	UVHP-ERV-04	2	32	34	2	2	32%		
36	2	HPS-1	201	HP-2	UVHP-ERV-04	2	30	32	2	2	30%		
37	1	HPS-1	SPECIAL	SPECIAL	Supply	30	0	0					

< >

Sheet1

+

Some Code Snippets (more available on request):

Main.py (only main function shown):

```
if __name__ == '__main__':
    """
    while True:
        try:
            number = int(input("How many different tags are there: "))
            break
        except ValueError:
            print("You have to enter numbers, try again")
    """

    schedule_list = []

    dataframe2 = pd.read_excel('input.xlsx', sheet_name='SCHEDULE')
    tupleScheduleList = list(dataframe2.itertuples(index=False,
name=None))

    for x in tupleScheduleList:
        tag = None
        model = None
        flow = None
        for y in range(3):
            if y == 0:
                tag = x[y]
            elif y == 1:
                model = x[y]
            elif y == 2:
                flow = x[y]
        schedule_list_object = Schedule(tag, model, flow)
        schedule_list.append(schedule_list_object)

    """
    for x in range(number):
        print("")
        print("Information for Tag " + str(x + 1))
        tag = input("Enter the tag: ")
        model = input("Enter the model number: ")
```

```

        flow = int(input("Enter the flow: "))
        list1.append(tuple((tag.upper(), model.upper(), flow)))
    """

    schedule_list.append(Schedule("SPECIAL", None, None))

    pipe_List = piping_chart()

    dataframe1 = pd.read_excel('book1.xlsx', sheet_name='TAKEOFF')

    tupleList = list(dataframe1.itertuples(index=False, name=None))

    floor_list = []
    riser_list = []

    for x in tupleList:
        floor = None
        riser = None
        suite = None
        schedule_object = None
        special = 0
        for y in range(4):
            if y == 0:
                floor = x[y]
            elif y == 1:
                riser = x[y]
                riser_list.append(riser)
            elif y == 2:
                suite = x[y]
                if str(suite).upper() == "SPECIAL":
                    special = 1
            elif y == 3:
                if special == 1:
                    schedule_object = Schedule("SPECIAL", "Supply",
int(x[y]))
                else:
                    schedule_object = find_schedule(x[y],
schedule_list)
                level = Levels(floor, riser.upper(), suite, schedule_object)
                floor_list.append(level)

        floor_list.sort(key=lambda a: a.floor)
        floor_list.sort(key=lambda a: a.riser)

    final_riser_list = []
    [final_riser_list.append(x) for x in riser_list if x not in
final_riser_list]

    while True:
        try:
            supply_level = int(input("What level is the supply run:
"))
            break
        except ValueError:
            print("You have to enter numbers, try again")

    while True:
        supply_loc = input("Is it at the Top/Bottom: ")
        if supply_loc.upper() == "TOP":
            break
        elif supply_loc.upper() == "BOTTOM":
            break
        else:
            print("Invalid input, try again")

    final_floor_list = table_fill(floor_list, final_riser_list,
supply_level, supply_loc)
    final_floor_list.sort(key=lambda a: a.floor, reverse=True)
    final_floor_list.sort(key=lambda a: a.riser)

    write_excel(final_floor_list, excel_index)

```

levels.py:

```
from schedule import Schedule

class Levels:

    def __init__(self, floor, riser, suite, schedule):
        self.floor = floor
        self.riser = riser
        self.suite = suite
        self.schedule = schedule
        self.bottomFlow = 0
        self.topFlow = 0
        self.topPipe = None
        self.bottomPipe = None
```

schedule.py:

```
class Schedule:

    def __init__(self, tag, model, flow):
        self.tag = tag
        self.model = model
        self.flow = flow
```