Ministry of Education and Science of Ukraine
Kharkiv National University of Radio Electronics

Ministry of Education and Science of Ukraine
Kharkiv National University of Radio Electronics

Laboratory work No1

Discipline: Algorithms and data structures

Done by
Student of group PZPI-21-4
Sorokin Vadym
Checked by
Senior Lecturer of the Department of PI
Oliynyk Olena Volodymyrivna

2021

Topic: Operations with sets
Goal: Create a program for of do operations with sets.

Code:
main.py:

```
import types

import string

import re

import sys


def add(a: list, b: list):  # 2

    return a + b

def crossing(a: list, b: list):  # 3

    return [x for x in a + b if x in a and x in b]

def minus(a: list, b: list):  # 3

    return [x for x in a if x not in b]

def anti(a: list):  # 1

    return [x for x in global_sets["u"] if x not in a]

def str_set(args: str):

    return str(global_sets[args])


procedures_order: list[dict[str, list[int,
types.FunctionType]]] = [

    {'~':[1, anti]}, {'+':[2, add]}, {'/': [2, crossing],
'-': [2, minus]}]

functions: dict[str, types.FunctionType] = {"print":
str_set}

global_sets: dict[str, list[str]] = {

    'u': list([str(x) for x in range(21)])

}
```

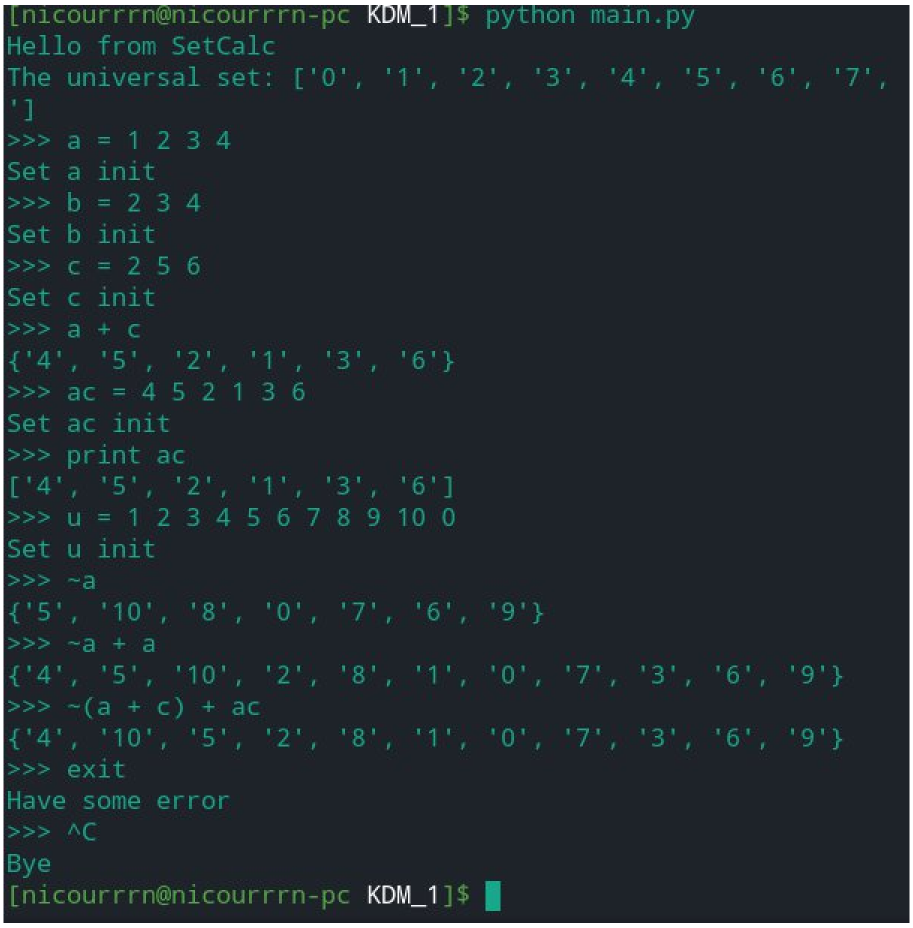
```
def brace_handline(line: str):
    regex = re.compile(r"\([^\)]*\)")
    parsed = []
    start_len = len(line)
    for i, c in enumerate(reversed(line)):
        if c == "(":
            pos = start_len - i - 1
            parsed.append(regex.findall(line[pos:])[0])
            line = line.replace(parsed[-1], str(len(parsed)
- 1))
    return parsed


def line_handler(line: str) -> list[str]:
    vars = re.findall(r"([\w~]+)", line)
    doings = ""
    for i in procedures_order[1:]:
        doings += "".join(i.keys())
    does = re.findall(r"([{0}]+)".format(doings), line)
    ret = [vars[0]]
    for i, j in zip(vars[1:], does):
        ret += [j, i]
    return ret


def procedure_handline(handlered_line: list) -> list:
    temp_calc = 0
    br = False
    for order in procedures_order:
        for _ in range(len(handlered_line) // 2 + 2):
            for i, el in enumerate(handlered_line):
                for proc in order.keys():
                    if proc in el:
                        index = str(temp_calc) + "t"
                        if (order[proc])[0] == 1:
                            handlered_line[i] = index
                            global_sets[index] =
order[proc][1](global_sets[el.replace(proc, "")])
                        elif (order[proc])[0] == 2:
                            global_sets[index] =
order[proc][1](global_sets[handlered_line[i - 1]],
global_sets[handlered_line[i + 1]])
                            handlered_line =
handlered_line[:i - 1] + [index] + handlered_line[i + 2:]
                        temp_calc += 1
                        br = True
                        break
                if br: break
            br = False
    return global_sets[str(temp_calc - 1) + "t"]


def get_data(line) -> str:
    if "=" in line:
        data = list(map(lambda x: x.strip(),
line.split("=")))
        global_sets[data[0]] = list(set(data[1].split("
")))
        return f"Set {data[0]} init"


    for fn in functions.keys():
        if fn in line:
            args = line.replace(fn, "").strip()
            return functions[fn](args)
    line = "(" + line + ")"

    line = line.replace(" ", "")

    for i, h in enumerate(brace_handline(line)):

        global_sets[str(i)] =
procedure_handline(list(line_handler(h)))

    else:

        return set(global_sets[str(i)])


if __name__ == "__main__":

    print("Hello from SetCalc")

    print(f"The universal set: {global_sets['u']}")


    while True:

        try:

            print(get_data(input(">>> ").strip()))

        except KeyboardInterrupt:

            print("\nBye")

            break

        except:

            print("Have some error")
```

gui.py:

```
from PyQt5.QtWidgets import QWidget, QPushButton, QLabel,
QVBoxLayout, QApplication, QLineEdit, QScrollArea, QFrame,
QScrollBar

from main import get_data



class Calc:

    def __init__(self):
        self.wind = QWidget()

        scroll = QScrollArea()

        scroll.setWidgetResizable(True)

        wind = QWidget()

        self.lay = QVBoxLayout()

        wind.setLayout(self.lay)

        scroll.setWidget(wind)

        self.lay.addWidget(QLabel("Hello from SetCalc\n\
The universal set: ['0', '1', '2', '3', '4', '5', '6', '7',
'8', '9', '10', '11', '12', '13', '14', '15', '16', '17',
'18', '19', '20']"))

        self.scroll = scroll

        self.input = QLineEdit()

        self.btn = QPushButton("Go!")

        self.btn.clicked.connect(self.hello)

        layaot = QVBoxLayout()

        layaot.addWidget(scroll)

        layaot.addWidget(self.input)

        layaot.addWidget(self.btn)

        self.wind.setMinimumSize(300, 400)

        self.wind.setLayout(layaot)

        self.wind.setWindowTitle("Calc")

        self.wind.move(300, 100)

        self.wind.show()


    def hello(self):

        line = "There are some err"

        try:

            line = get_data(self.input.text().strip())

        except:
                pass

            self.input.setText("")

            l = QLabel(str(line))

            self.lay.insertWidget(0, l)


app = QApplication([])

c = Calc()

app.exec()
```

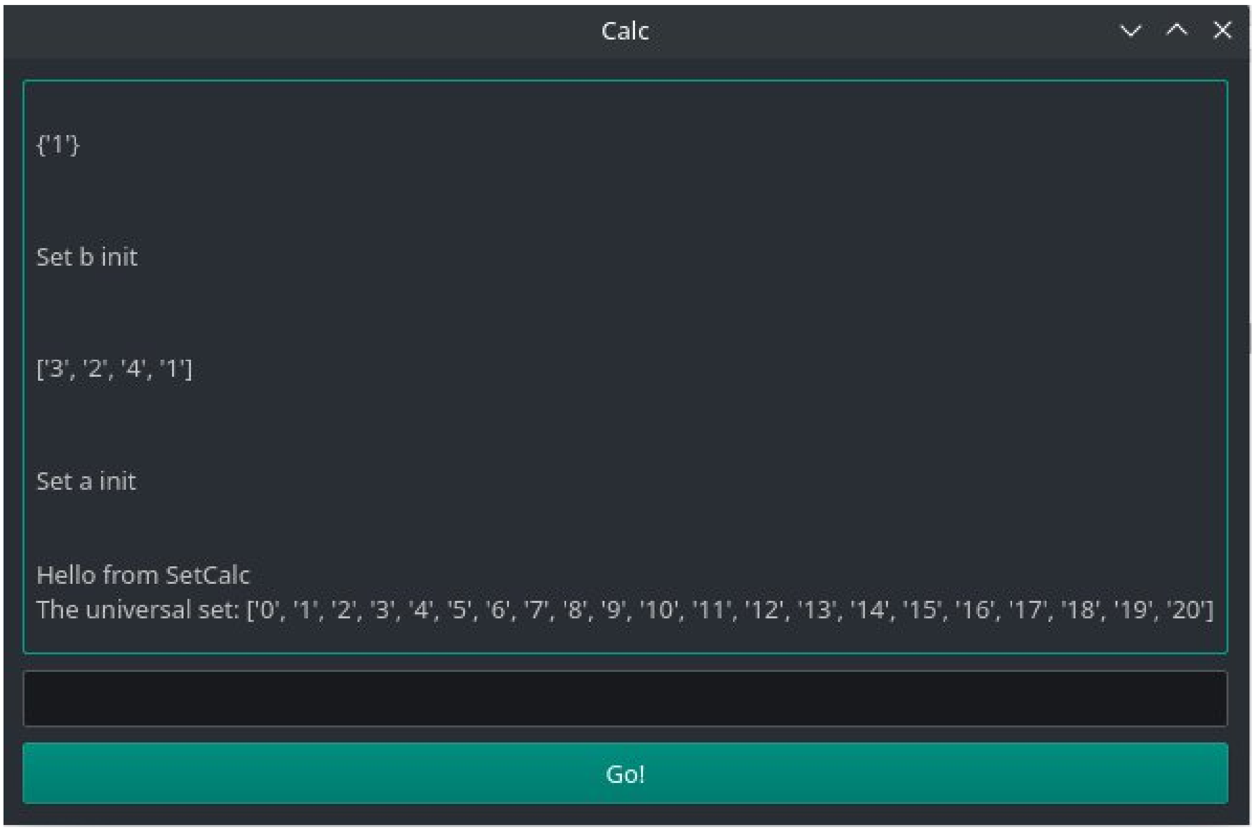
App is wrote on Python 3.9. For run application with GUI you need to install Python package named PyQt5. You can also download code from github: *git@github.com:nicourn/SetCalc.git*

Screenshots:



```
[nicourrrn@nicourrrn-pc KDM_1]$ python main.py
Hello from SetCalc
The universal set: ['0', '1', '2', '3', '4', '5', '6', '7',
']
>>> a = 1 2 3 4
Set a init
>>> b = 2 3 4
Set b init
>>> c = 2 5 6
Set c init
>>> a + c
{'4', '5', '2', '1', '3', '6'}
>>> ac = 4 5 2 1 3 6
Set ac init
>>> print ac
['4', '5', '2', '1', '3', '6']
>>> u = 1 2 3 4 5 6 7 8 9 10 0
Set u init
>>> ~a
{'5', '10', '8', '0', '7', '6', '9'}
>>> ~a + a
{'4', '5', '10', '2', '8', '1', '0', '7', '3', '6', '9'}
>>> ~(a + c) + ac
{'4', '10', '5', '2', '8', '1', '0', '7', '3', '6', '9'}
>>> exit
Have some error
>>> ^C
Bye
[nicourrrn@nicourrrn-pc KDM_1]$
```

1.1.   Run from console



```
{'1'}


Set b init


['3', '2', '4', '1']


Set a init


Hello from SetCalc
The universal set: ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20']
```

1.2 GUI application

Conclusion: Learned how to create complicated calculators. Implemented algorithms using the Python programming language.