Python training - lab 14

Exceptions

- an event, which occurs during the execution of a program that disrupts the normal flow of the program's instructions
- triggered automatically on errors or by the code

```
l = ['a', 'b']
print(l[4])
# IndexError: list index out of range

n = input('Enter the number:')
print(n ** 2)
# TypeError: unsupported operand type(s) for ** or pow():
'str' and 'int'
```

- exceptions hierarchy: https://docs.python.org/3/library/exceptions.html#exception-hierarchy

Exceptions handling

```
try:
   l = ['a', 'b']
   print(l[4])
   print('one more line..')
except:
    print("Exception caught!")
else:
    print("No exception caught!")
finally:
    print("Finally done!")

print('Program continues..')
```

Exceptions handling

```python
try:
    pass
except LookupError:
    print('LookupError caught')
except IndexError as ex:
    print(type(ex), ex)
except (IndexError, KeyError) as ex:
    print(type(ex), ex)
except Exception:
    print('Everything..')
else:
    print("No exception caught!")
finally:
    print("Finally...")

print('Will this print anything?')
```

Custom exceptions

```
class MyCustomException(LookupError):
   pass

# raise - raises (throws) an exception

raise MyCustomException('some text')

raise IndexError
```

Static class members

```
class A:

    instances = 0

    def __init__(self, x):
        self.x = x
        A.instances += 1

    def __del__(self):
        A.instances -= 1

    @staticmethod
    def show_instances():
        print(f'There are {A.instances} instances so far..')


a = A(7)
b = A(8)
A.instances
A.show_instances()
```