Лабораторна робота 3
Варіант 5
Ничипорука Дмитра

git: https://github.com/Djonny-svoboden/OOP1Semester/tree/main/lab3

1)Абстрактний клас

```
###################################
class Person(metaclass=ABCMeta):  # superclass
    @abstractmethod
    def create_password(self):
        pass

    @abstractmethod
    def change_password(self):
        pass

    @abstractmethod
    def log_in(self):
        pass

    def pass_check(self):
        if not self._password:
            print("U must create password")
            Employer.create_password(self)
            return True
        else:
            return False


###################################
```

2)Класс працівника

```python
class Employer(Person):  # subclass
    def __init__(self, name, password):
        self.name = name
        self._password = password
        if Employer.pass_check(self):
            Employer.log_in(self)

    def __str__(self):
        return self.name

    def create_password(self):
        print("Type ur password")
        self._password = input()
        print("Success")

    def change_password(self):
        check_pass = input()
        if check_pass == self._password:
            self._password = input()
            print("U successfully changed password ")
        else:
            print("Ur not employer")
        pass

    def log_in(self):
        print("Password:")
        new_pass = input()
        if new_pass != self._password:
            print("Ur not employer")
        else:
            print("Welcome", self.name)
```

3)Клас адміністратора

```
###################################
class Administrator(Person):  # subclass
    def __init__(self, name, password, secret):
        self.name = name
        self._password = password
        self._secret = secret
        if Employer.pass_check(self):
            Employer.log_in(self)

    def __str__(self):
        return self.name

    def create_password(self):
        print("Type ur password")
        self._password = input()
        print("Success")
        print("Type ur Secret")
        self._secret = input()
        print("Success")

    def change_password(self):
        check_pass = input()
        check_secret = input()
        if check_pass == self._password and check_secret == self._secret:
            self._password = input()
            print("U successfully changed password ")
        else:
            print("Ur not employer")
        pass

    def log_in(self):
        print("Password:")
        new_pass = input()
        print("Secret:")
        new_secret = input()
        if new_pass != self._password or new_secret != self._secret:
            print("Ur not administrator")
        else:
            print("Welcome", self.name)


###################################
```

4)Клас-контейнер

```
class Persons(Collection):  # class container
    def __init__(self):
        self.list = []

    def add_employer(self, employer):
        if isinstance(employer, Employer):
            self.list.append(employer)

    def add_admin(self, administrator):
        if isinstance(administrator, Administrator):
            self.list.append(administrator)

    def del_person(self, num):
        if num < len(self.list):
            rem = self.list.pop(num - 1)
            print

    def __contains__(self, item):
        return (item in self.list)

    def __len__(self):
        return len(self.list)

    def __getitem__(self, item):
        if isinstance(item, int):
            if item < len(self.list) and item > len(self.list) * -1:
                return self.list[item]
            else:
                raise IndexError()

    def __iter__(self):
        return self.list.__iter__()
```

5)Виклик

```
if __name__ == "__main__":
    ### employers
    emp1 = Employer("NEWNAME1", "pass1")
    emp2 = Employer("NEWNAME2", "pass1")
    emp3 = Employer("NEWNAME3", "pass1")
    emp4 = Employer("NEWNAME4", "pass1")

    emp1.log_in()
    ### administrator
    adm1 = Administrator("NewName1", "pass1", "answer1")
    adm2 = Administrator("NewName2", "pass2", "answer2")
    adm3 = Administrator("NewName3", "pass3", "answer3")

    adm1.log_in()

    ### container
    # ec.change_password()
    p = Persons()
    p.add_employer(emp1)
    p.add_employer(emp2)
    p.add_employer(emp3)
    p.add_employer(emp4)
    p.add_admin(adm1)
    p.add_admin(adm2)
    p.add_admin(adm3)
    print(isinstance(p, collections.abc.Container))
    for i in p:
        print(i)
    # print(p.__getitem__(0))
```

3

Висновок: я ознайомився з поняттям абстрактного класу в Пайтон та навчився їх створювати