|  |
| --- |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** |
| Институт кибербезопасности и цифровых технологий |
| Кафедра КБ-2 «Прикладные информационные технологии» |

**ПРАКТИЧЕСКАЯ РАБОТА №6 по дисциплине**

**«Технологии и методы программирования»**

***(наименование дисциплины)***

**Тема практической работы   
«Итератор. Посетитель»**

Студент группы: БИСО-01-20 Давыдова Д.А.

*(шифр учебной группы) (Фамилия И.О)*

Преподаватель: Лесько С.А.

*(Фамилия И.О)*

Москва, 2023

Практическая работа №6

[**Инверсия управления** 3](#_Toc135827313)

[**Заместитель** 4](#_Toc135827314)

[**Компановщик** 5](#_Toc135827315)

**Инверсия управления**

Листинг программы:

people = []

numbers=[]

def hello\_people():

    for person in people:

        print(f"Привет, {person}.")

def calculation(sum=0):

    for num in numbers:

        sum=num+sum

    print(sum)

people.append("мама")

people.append("папа")

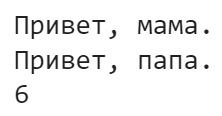
hello\_people()

numbers.append(4)

numbers.append(2)

calculation()

Результат работы алгоритма:



**Заместитель**

Листинг программы:

from abc import ABC, abstractmethod

class PasswordItem:

    def \_\_init\_\_(self, password):

        self.password = password

    def get(self):

        return self.password

class Target(ABC):

    @abstractmethod

    def ex(self) -> None:

        pass

class Real(Target):

    def ex(self) -> None:

        print("Настоящая часть кода запущена...\n7 \* 7 =", 7\*7)

class Proxy(Target):

    def \_\_init\_\_(self, real\_target: Real) -> None:

        self.\_real\_target = real\_target

    def ex(self) -> None:

        if self.access():

            self.\_real\_target.ex()

    def access(self) -> bool:

        realpassword = 12345

        print("Proxy: Проверка доступа")

        if realpassword == password.get():

            return True

        else:

            print("В доступе отказано")

            return False

def client(target: Target) -> None:

    target.ex()

if \_\_name\_\_ == "\_\_main\_\_":

    password = PasswordItem(111111)

    print(password.get())

    print("Запуск напрямую, без Proxy:")

    real\_target = Real()

    client(real\_target)

    print("\nЗапуск с Proxy и неправильным паролем:")

    proxy = Proxy(real\_target)

    client(proxy)

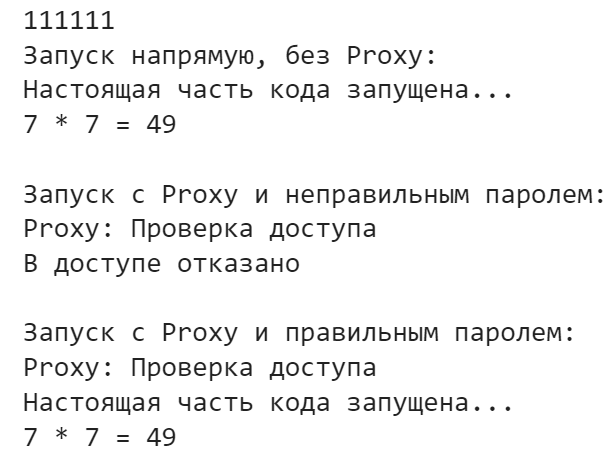
    password = PasswordItem(12345)

    print("\nЗапуск с Proxy и правильным паролем:")

    proxy = Proxy(real\_target)

    client(proxy)

Результат работы алгоритма:



**Компановщик**

Листинг программы:

class Leaf:

    def \_\_init\_\_(self, number, \*args):

        self.number = number

        self.position = args[0]

    def show(self):

        print("\t", end="")

        print(self.position)

    def showNum(self):

        print("\t", end="")

        print(self.number)

class Element:

    def \_\_init\_\_(self, number, \*args):

        self.number = number

        self.position = args[0]

        self.children = []

    def add(self, child):

        self.children.append(child)

    def remove(self, child):

        self.children.remove(child)

    def show(self):

        print(self.position)

        for child in self.children:

            print("\t", end="")

            child.show()

    def showNum(self):

        print(self.number)

        for child in self.children:

            print("\t", end="")

            child.showNum()

if \_\_name\_\_ == "\_\_main\_\_":

    Hi = Element("A1", "Генерал")

    Item1 = Element("B1", "Майор 1")

    Item2 = Element("B2", "Майор 2")

    Item3 = Element("B3", "Майор 3")

    Item11 = Leaf("C1", "Рядовой 11")

    Item12 = Leaf("C2", "Рядовой 12")

    Item21 = Leaf("C3", "Рядовой 21")

    Item22 = Leaf("C4", "Рядовой 22")

    Item31 = Leaf("C5", "Рядовой 31")

    Item32 = Leaf("C6", "Рядовой 32")

    Item1.add(Item11)

    Item1.add(Item12)

    Item2.add(Item21)

    Item2.add(Item22)

    Item3.add(Item31)

    Item3.add(Item32)

    Hi.add(Item1)

    Hi.add(Item2)

    Hi.add(Item3)

    Hi.show()

    print("Номера:")

    Hi.showNum()

Результат работы алгоритма:

