

1.

# Модуль biblioteca.py

class Biblioteca:

def \_\_init\_\_(self, name):

self.name = name

# Модуль язык\_программирования.py

class ЯзыкПрограммирования:

def \_\_init\_\_(self, name):

self.name = name

self.biblioteci = []

def добавить\_библиотеку(self, biblioteca):

self.biblioteci.append(biblioteca)

# Модуль main.py

from biblioteca import Biblioteca

from язык\_программирования import ЯзыкПрограммирования

def main():

# Создаем объекты библиотек

biblioteca1 = Biblioteca("math")

biblioteca2 = Biblioteca("random")

biblioteca3 = Biblioteca("datetime")

biblioteca4 = Biblioteca("React")

biblioteca5 = Biblioteca("devExtreme")

biblioteca6 = Biblioteca("<iostream>")

# Создаем объекты языков программирования

язык1 = ЯзыкПрограммирования("Python")

язык2 = ЯзыкПрограммирования("JavaScript")

язык3 = ЯзыкПрограммирования("C++")

```

# Связываем языки программирования с библиотеками
язык1.добавить_библиотеку(biblioteca1)
язык1.добавить_библиотеку(biblioteca2)
язык1.добавить_библиотеку(biblioteca3)
язык2.добавить_библиотеку(biblioteca4)
язык2.добавить_библиотеку(biblioteca5)
язык3.добавить_библиотеку(biblioteca6)

# Выполняем различные операции
вывод_языков_с_ов(язык1, язык2, язык3)
вывод_языков_с_средним_количеством_библиотек(язык1, язык2, язык3)
вывод_библиотек_с_названием_на_д(язык1, язык2, язык3)

def вывод_языков_с_ов(*языки):
    языки_с_ов = [(язык.name, [biblioteca.name for biblioteca in язык.bibliotecai])
                   for язык in языки if язык.name.endswith("n")]
    print(языки_с_ов)

def вывод_языков_с_средним_количеством_библиотек(*языки):
    языки_количество_библиотек = [(язык.name, len(язык.bibliotecai))
                                    for язык in языки]
    языки_количество_библиотек.sort(key=lambda x: x[1])
    print(языки_количество_библиотек)

def вывод_библиотек_с_названием_на_д(*языки):
    библиотеки_с_A = [(biblioteca.name, [язык.name for язык in языки])
                      for biblioteca in [biblioteca1, biblioteca2, biblioteca3, biblioteca4, biblioteca5, biblioteca6] if
                      biblioteca.name.startswith("d")]
    print(библиотеки_с_A)

if __name__ == "__main__":
    main()

```

2.

```
import unittest

from biblioteca import Biblioteca

from язык_программирования import ЯзыкПрограммирования

class TestMain(unittest.TestCase):

    def setUp(self):

        # Настройка начальных условий для каждого теста

        self.biblioteca1 = Biblioteca("math")

        self.biblioteca2 = Biblioteca("random")

        self.язык1 = ЯзыкПрограммирования("Python")

        self.язык2 = ЯзыкПрограммирования("JavaScript")

    def test_добавить_библиотеку(self):

        self.язык1.добавить_библиотеку(self.biblioteca1)

        self.assertEqual(len(self.язык1.bibliotec1), 1)

        self.assertEqual(self.язык1.bibliotec1[0].name, "math")

    def test_языки_с_ов(self):

        языки_с_ов = [(язык.name, [biblioteca.name for biblioteca in язык.bibliotec1])

                        for язык in [self.язык1, self.язык2] if язык.name.endswith("n")]

        self.assertEqual(языки_с_ов, [("Python", []), ("JavaScript", [])])

    def test_библиотеки_с_названием_на_д(self):

        self.язык1.добавить_библиотеку(self.biblioteca1)

        self.язык1.добавить_библиотеку(self.biblioteca2)

        библиотеки_с_А = [(biblioteca.name, [язык.name for язык in [self.язык1, self.язык2]])

                            for biblioteca in [self.biblioteca1, self.biblioteca2] if biblioteca.name.startswith("d")]

        self.assertEqual(библиотеки_с_А, [])
```

```
if __name__ == '__main__':  
    unittest.main()
```