

**Московский государственный технический  
университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»  
Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Парадигмы и конструкции языков программирования»

Отчет по рубежному контролю №1

Вариант А22

Выполнил:  
Студент группы ИУ5-31Б  
Чуботов Лука

Проверил:  
Преподаватель каф. ИУ5  
Гапанюк Ю. Е.

Москва, 2025 г.

## Листинг программы:

### main.py:

```
from typing import List
from typing import Tuple
from typing import Callable
from operator import itemgetter
from dataclasses import dataclass

@dataclass
class Library:
    lib_id: int
    name: str
    size: int
    lang_id: int

@dataclass
class ProgrammingLanguage:
    lang_id: int
    name: str

@dataclass
class LibraryLanguage:
    lib_id: int
    lang_id: int

def create_languages() -> List[ProgrammingLanguage]:
    return [
        ProgrammingLanguage(1, 'Python'),
        ProgrammingLanguage(2, 'JavaScript'),
        ProgrammingLanguage(3, 'C++'),
        ProgrammingLanguage(4, 'Java'),
        ProgrammingLanguage(5, 'TypeScript'),
    ]

def create_libraries() -> List[Library]:
```

```

    return [
        Library(1, 'NumPy', 45, 1),
        Library(2, 'Pandas', 85, 1),
        Library(3, 'Django', 120, 1),
        Library(4, 'React', 180, 2),
        Library(5, 'Vue.js', 95, 2),
        Library(6, 'Boost', 250, 3),
        Library(7, 'Qt', 200, 3),
        Library(8, 'Spring', 150, 4),
    ]

def create_library_language_relations() -> List[LibraryLanguage]:
    return [
        LibraryLanguage(1, 1),
        LibraryLanguage(2, 1),
        LibraryLanguage(3, 1),
        LibraryLanguage(4, 2),
        LibraryLanguage(4, 5),
        LibraryLanguage(5, 2),
        LibraryLanguage(5, 5),
        LibraryLanguage(6, 3),
        LibraryLanguage(7, 3),
        LibraryLanguage(8, 4),
    ]

def print_table(data: List[Tuple], title: str, headers: List[str], column_width: int = 20) -> None:
    if not data:
        print(f'{title}: Нет данных\n')
        return

    print(f'\n{title}\n')
    header_format = f''.join([f'{{: <{column_width}}}' for _ in headers])
    print(header_format.format(*headers))
    for row in data:
        print(header_format.format(*row))

def task_1(languages: List[ProgrammingLanguage], libraries: List[Library]) -> List[Tuple[str, str, int]]:
    result = [

```

```

        (lang.name, lib.name, lib.size)

        for lang in languages

        for lib in libraries

        if lib.lang_id == lang.lang_id

    ]

return sorted(result, key=itemgetter(0))
}

def task_2(languages: List[ProgrammingLanguage], libraries: List[Library]) -> List[Tuple[str, int]]:

    result = []

    for lang in languages:

        filtered_libs = list(filter(lambda lib: lib.lang_id == lang.lang_id, libraries))

        if filtered_libs:

            total_size = sum([lib.size for lib in filtered_libs])

            result.append((lang.name, total_size))

    return sorted(result, key=itemgetter(1), reverse=True)

}

def task_3(languages: List[ProgrammingLanguage], libraries: List[Library], relations:
List[LibraryLanguage], condition: Callable[[str], bool]) -> List[Tuple[str, str]]:

    filtered_langs = list(filter(lambda lang: condition(lang.name), languages))

    result = []

    for lang in filtered_langs:

        lib_ids = [
            rel.lib_id
            for rel in relations
            if rel.lang_id == lang.lang_id
        ]

        lib_names = [
            lib.name
            for lib in libraries
            if lib.lib_id in lib_ids
        ]

```

```

        for lib_name in lib_names:
            result.append((lang.name, lib_name))

    return sorted(result, key=itemgetter(0, 1))

def main() -> None:
    languages = create_languages()
    libraries = create_libraries()

    print_table(task_1(languages, libraries), "Задание 1", ["Язык", "Библиотека", "Размер (МБ)"])
    print_table(task_2(languages, libraries), "Задание 2", ["Язык", "Суммарный размер (МБ)"])

    relations = create_library_language_relations()
    print_table(task_3(languages, libraries, relations, lambda name: "Python" in name or "Type" in name),
    "Задание 3", ["Язык", "Библиотека"])

if __name__ == '__main__':
    main()

```

## Результат выполнения:

### Задание 1

Язык	Библиотека	Размер (МБ)
C++	Boost	250
C++	Qt	200
Java	Spring	150
JavaScript	React	180
JavaScript	Vue.js	95
Python	NumPy	45
Python	Pandas	85
Python	Django	120

### Задание 2

Язык	Суммарный размер (МБ)
C++	450
JavaScript	275
Python	250
Java	150

### Задание 3

Язык	Библиотека
Python	Django
Python	NumPy
Python	Pandas
TypeScript	React
TypeScript	Vue.js