

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
им. Н.Э. Баумана

Факультет “Информатика и системы управления”
Кафедра “Системы обработки информации и управления”



Дисциплина “Парадигмы и конструкции языков программирования”

Отчет по рубежному контролю №1

Выполнил:
Студент группы ИУ5-34Б
Шерстеникин Н.С.
Преподаватель:
Гапанюк Ю.Е.

Москва 2025

```

class Stroki:
    def __init__(self, Id, Content, CharCount, TableId):
        self.Id = Id
        self.Content = Content
        self.CharCount = CharCount
        self.TableId = TableId

    def __repr__(self):
        return f"Stroki(Id={self.Id}, Content='{self.Content}',
CharCount={self.CharCount}, TableId={self.TableId})"

class Table:
    def __init__(self, Id, Name):
        self.Id = Id
        self.Name = Name

    def __repr__(self):
        return f"Table(Id={self.Id}, Name='{self.Name}')"

class TableStroka:
    def __init__(self, Id, StringId, TableId):
        self.Id = Id
        self.StringId = StringId
        self.TableId = TableId

    def __repr__(self):
        return f"TableStroka(Id={self.Id}, StringId={self.StringId},
TableId={self.TableId})"

StrokiList = [
    Stroki(1, "бла-бла", 7, 1),
    Stroki(2, "барбер", 6, 2),
    Stroki(3, "ода", 3, 3),
    Stroki(4, "гильгамеш", 9, 3)
]

Tables = [
    Table(1, "текст"),
    Table(2, "мало текста"),
    Table(3, "много текста")
]

TableStroki = [
    TableStroka(1, 1, 1),
    TableStroka(2, 2, 2),

```

```

    TableStroka(3, 3, 3),
    TableStroka(4, 4, 3),
    TableStroka(5, 3, 2) # Добавил эту строку сюда, чтобы показать связь
многие-ко-многим
]

```

```

def RequestOne(StrokiList, Tables):

```

```

    """

```

```

    Запрос 1: Строки, начинающиеся с "б" и их таблицы
    """

```

```


```

```

    StringsWithB = [S for S in StrokiList if S.Content.startswith('б')]

```

```

    Result = list(map(lambda S: {

```

```

        'StringContent': S.Content,

```

```

        'TableName': next((T.Name for T in Tables if T.Id == S.TableId),

```

```

None),

```

```

        'StringId': S.Id,

```

```

        'TableId': S.TableId

```

```

    }, StringsWithB))

```

```

}, StringsWithB))

```

```

    'StringId': S.Id,

```

```

    'TableId': S.TableId

```

```

}, StringsWithB))

```

```

return Result

```

```

def RequestTwo(StrokiList, Tables):

```

```

    """

```

```

    Запрос 2: Минимальные количества символов в строках таблиц,
отсортированные по возрастанию
    """

```

```


```

```

    MinCharsPerTable = {

```

```

        TableObj.Name: min(S.CharCount for S in StrokiList if S.TableId ==
TableObj.Id)

```

```

        for TableObj in Tables

```

```

        if any(S.TableId == TableObj.Id for S in StrokiList)

```

```

    }

```

```

    SortedResult = sorted(MinCharsPerTable.items(), key=lambda X: X[1])

```

```

    return SortedResult

```

```

def RequestThree(StrokiList, Tables, TableStroki):

```

```

    """

```

```

    Запрос 3: Все строки и их таблицы (используя связь многие-ко-
многим), отсортированные по ID строки
    """

```

```


```

```

Result = [
    {
        'StringId': Stroka.Id,
        'StringContent': Stroka.Content,
        'TableName': TableObj.Name,
        'TableId': TableObj.Id
    }
    for Ts in TableStroki
    for Stroka in StrokiList if Stroka.Id == Ts.StringId
    for TableObj in Tables if TableObj.Id == Ts.TableId
]
SortedResult = sorted(Result, key=lambda X: X['StringId'])
return SortedResult

def Main():
    print("\n" + "ЗАПРОС 1: Строки, начинающиеся с 'б' и их таблицы")

    Result1 = RequestOne(StrokiList, Tables)
    for Item in Result1:
        print(f"Строка: '{Item['StringContent']}', Таблица: '{Item['TableName']}'")

    print("\n" + "ЗАПРОС 2: Минимальные количества символов в строках таблиц")

    Result2 = RequestTwo(StrokiList, Tables)
    for TableName, MinChars in Result2:
        print(f"Таблица '{TableName}': {MinChars} символов")

    print("\n" + "ЗАПРОС 3: Все строки и их таблицы (отсортировано по ID строки)")

    Result3 = RequestThree(StrokiList, Tables, TableStroki)
    for Item in Result3:
        print(f"ID строки: {Item['StringId']}, Содержание: '{Item['StringContent']}', Таблица: '{Item['TableName']}'")

Main()

```

Работа кода:

ЗАПРОС 1: Строки, начинающиеся с 'б' и их таблицы

Строка: 'бла-бла', Таблица: 'текст'

Строка: 'барбер', Таблица: 'мало текста'

ЗАПРОС 2: Минимальные количества символов в строках таблиц

Таблица 'много текста': 3 символов

Таблица 'мало текста': 6 символов

Таблица 'текст': 7 символов

ЗАПРОС 3: Все строки и их таблицы (отсортировано по ID строки)

ID строки: 1, Содержание: 'бла-бла', Таблица: 'текст'

ID строки: 2, Содержание: 'барбер', Таблица: 'мало текста'

ID строки: 3, Содержание: 'ода', Таблица: 'много текста'

ID строки: 3, Содержание: 'ода', Таблица: 'мало текста'

ID строки: 4, Содержание: 'гильгамеш', Таблица: 'много текста'