

# Рубежный контроль №1 по курсу ПиК ЯП

## Вариант Г.

1. «База данных» и «Хранимая процедура» связаны соотношением один-ко-многим. Выведите список всех баз данных, у которых в названии есть нечетное число, и список хранимых в них процедур.
2. «База данных» и «Хранимая процедура» связаны соотношением один-ко-многим. Выведите список баз данных с максимальным весом процедур в каждой базе данных, отсортированный по максимальному весу.
3. «База данных» и «Хранимая процедура» связаны соотношением многие-ко-многим. Выведите список всех связанных баз данных и процедур, отсортированный по базам данных, сортировка по процедурам произвольная.

№ варианта	Класс 1	Класс 2
34	Хранимая процедура	База данных

```
class Procedure:
```

```
    def __init__(self, id, name, size, db_id):
        self.id = id
        self.name = name
        self.size = size
        self.db_id = db_id
```

```
class DataBase:
```

```
    def __init__(self, id, name):
        self.id = id
        self.name = name
```

```
class ProcedureDB:
```

```
    def __init__(self, db_id, procedure_id):
        self.db_id = db_id
        self.procedure_id = procedure_id
```

```
dbs = [
```

```
    DataBase(1, 'db1'),
```

```
        DataBase(2, 'db2'),
        DataBase(3, 'db3')
]

procedures = [
    Procedure(1, 'p1', 16, 1),
    Procedure(2, 'p2', 32, 1),
    Procedure(3, 'p3', 64, 2),
    Procedure(4, 'p4', 128, 2),
    Procedure(5, 'p5', 256, 3),
    Procedure(6, 'p6', 512, 3)
]
```

```
dbs_to_procedures = [
    ProcedureDB(1, 1),
    ProcedureDB(1, 2),
    ProcedureDB(1, 3),

    ProcedureDB(2, 4),
    ProcedureDB(2, 5),
    ProcedureDB(2, 6),

    ProcedureDB(3, 1),
    ProcedureDB(3, 6)
]
```

```
def main():

    print("Здание №1")
    dict1={}
```

```

for db in dbs:
    if int(db.name[-1])%2==1:
        dict1[db.name] = [procedure.name for procedure in
procedures if procedure.db_id==db.id]

for db, procedure in dict1.items():
    print(db, ': ', end=' ')
    print(*procedure, sep=', ')

print("Здание №2")
dict2={}
for db in dbs:
    dict2[db.name] = max([procedure.size for procedure in
procedures if procedure.db_id==db.id])
    for db, size in sorted(dict2.items(), key=lambda x: x[1],
reverse=True):
        print(db, ': ', size)

print("Здание №3")
dict3={}
for db in dbs:
    a=[]
    for con in dbs_to_procedures:
        if db.id==con.db_id:
            for procedure in procedures:
                if procedure.id == con.procedure_id:
                    a.append(procedure.name)
    if len(a)>0:
        dict3[db.name] = a
for db, procedure in dict3.items():
    print(db, ': ', end=' ')

```

```
print(*procedure, sep=', ')
```

```
main()
```

Здание №1

db1 : p1, p2

db3 : p5, p6

Здание №2

db3 : 512

db2 : 128

db1 : 32

Здание №3

db1 : p1, p2, p3

db2 : p4, p5, p6

db3 : p1, p6