

# Рубежный контроль №1

Вариант 17 — Дирижёр / Оркестр (Вариант Г)

## Текст программы

```
from operator import itemgetter
from reportlab.platypus import SimpleDocTemplate, Paragraph, Spacer, Preformatted
from reportlab.lib.styles import getSampleStyleSheet, ParagraphStyle
from reportlab.lib.pagesizes import A4
from reportlab.pdfbase import pdfmetrics
from reportlab.pdfbase.ttfonts import TTFont
from io import StringIO
import sys
import inspect

# Регистрируем шрифт с поддержкой кириллицы
pdfmetrics.registerFont(TTFont('DejaVu', 'DejaVuSans.ttf'))

class Conductor:
    def __init__(self, id, fio, fee, orch_id):
        self.id = id
        self.fio = fio
        self.fee = fee
        self.orch_id = orch_id

class Orchestra:
    def __init__(self, id, name):
        self.id = id
        self.name = name

class ConductorOrchestra:
    def __init__(self, orch_id, cond_id):
        self.orch_id = orch_id
        self.cond_id = cond_id

orchestras = [
    Orchestra(1, 'Академический симфонический оркестр'),
    Orchestra(2, 'Ансамбль камерной музыки'),
    Orchestra(3, 'Большой оркестр'),
    Orchestra(4, 'Альтернативный джаз оркестр'),
]
conductors = [
    Conductor(1, 'Иванов', 120000, 1),
    Conductor(2, 'Петров', 90000, 1),
    Conductor(3, 'Сидоров', 150000, 2),
    Conductor(4, 'Алексеев', 110000, 3),
    Conductor(5, 'Орлов', 130000, 4),
]
conductors_orchestras = [
    ConductorOrchestra(1, 1),
    ConductorOrchestra(1, 2),
    ConductorOrchestra(2, 3),
```

```

ConductorOrchestra(3, 4),
ConductorOrchestra(4, 5),
ConductorOrchestra(4, 1),
]

def main():
    output = []

    one_to_many = [
        (c.fio, c.fee, o.name)
        for o in orchestras
        for c in conductors
        if c.orch_id == o.id
    ]

    many_to_many = [
        (c.fio, c.fee, o.name)
        for o in orchestras
        for co in conductors_orchestras
        for c in conductors
        if o.id == co.orch_id and c.id == co.cond_id
    ]

    output.append("Задание Г1")
    res1 = {}
    for o in orchestras:
        if o.name.startswith('A'):
            res1[o.name] = [fio for fio, _, name in one_to_many if name == o.name]
    output.append(str(res1))

    output.append("\nЗадание Г2")
    res2 = []
    for o in orchestras:
        fees = [fee for _, fee, name in one_to_many if name == o.name]
        if fees:
            res2.append((o.name, max(fees)))
    output.append(str(sorted(res2, key=itemgetter(1), reverse=True)))

    output.append("\nЗадание Г3")
    output.append(str(sorted(many_to_many, key=itemgetter(2)))))

    return "\n".join(output)

def create_pdf(program_text, result_text):
    doc = SimpleDocTemplate("RK1_Variant17_G.pdf", pagesize=A4)

    styles = getSampleStyleSheet()

    styles["Normal"].fontName = "DejaVu"
    styles["Title"].fontName = "DejaVu"
    styles["Heading2"].fontName = "DejaVu"

    code_style = ParagraphStyle(
        'Code',
        fontName='DejaVu',
        fontSize=9,
        leading=11
    )

    elements = []

```

```
elements.append(Paragraph("Рубежный контроль №1", styles["Title"]))
elements.appendSpacer(1, 12)
elements.append(Paragraph("Вариант 17 — Дирижёр / Оркестр (Вариант Г)", styles["Normal"]))
elements.appendSpacer(1, 12)

elements.append(Paragraph("Текст программы", styles["Heading2"]))
elements.append(Preformatted(program_text, code_style))
elements.appendSpacer(1, 12)

elements.append(Paragraph("Результаты выполнения", styles["Heading2"]))
elements.append(Preformatted(result_text, code_style))

doc.build(elements)

if __name__ == "__main__":
    buffer = StringIO()
    sys.stdout = buffer

    result = main()

    sys.stdout = sys.__stdout__

    program_text = inspect.getsource(sys.modules[__name__])
    create_pdf(program_text, result)

    print("PDF создан: RK1_Variant17_G.pdf")
```

## Результаты выполнения

Задание Г1

{'Академический симфонический оркестр': ['Иванов', 'Петров'], 'Ансамбль камерной музыки': ['Сидоров'], 'А

Задание Г2

[('Ансамбль камерной музыки', 150000), ('Альтернативный джаз оркестр', 130000), ('Академический симфони

Задание Г3

[('Иванов', 120000, 'Академический симфонический оркестр'), ('Петров', 90000, 'Академический симфониче