

**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»  
Рубежный контроль №2  
Вариант Г №1

Выполнил:  
Студент ИУ5-34Б  
Беккиев Р. И.  
Подпись и дата:

Проверил:  
Преподаватель каф. ИУ5  
Гапанюк Ю. Е.  
Подпись и дата:

Москва, 2023 г.

## Задание

### Условия рубежного контроля №2 по курсу ПнК ЯП

Рубежный контроль представляет собой разработку тестов на языке Python.

- 1) Проведите рефакторинг текста программы рубежного контроля №1 таким образом, чтобы он был пригоден для модульного тестирования.
- 2) Для текста программы рубежного контроля №1 создайте модульные тесты с применением TDD - фреймворка (3 теста).

## Текст программы

### main.py:

```
# main.py

class Group:
    def __init__(self, group_id, name):
        self.group_id = group_id
        self.name = name
        self.students = []

    def add_student(self, student):
        self.students.append(student)

class Student:
    def __init__(self, student_id, name, birth_year, group):
        self.student_id = student_id
        self.name = name
        self.birth_year = birth_year
        self.group = group

    def __str__(self):
        return f"{self.name} ({self.birth_year})"

class ManyToMany:
    def __init__(self, student_id, group_id):
        self.student_id = student_id
        self.group_id = group_id

# Создаем тестовые данные
group1 = Group(1, "А1")
group2 = Group(2, "Б1")

student1 = Student(1, "Иванов", 1998, group1)
student2 = Student(2, "Петров", 1997, group1)
student3 = Student(3, "Сидоров", 1999, group1)
student4 = Student(4, "Иваненко", 1996, group2)
student5 = Student(5, "Артемов", 1999, group2)

group1.add_student(student1)
group1.add_student(student2)
group1.add_student(student3)
group2.add_student(student4)
group2.add_student(student5)

groups = [group1, group2]
students = [student1, student2, student3, student4, student5]

many_to_many_list = [
    ManyToMany(1, 1),
```

```

        ManyToMany(2, 1),
        ManyToMany(3, 1),
        ManyToMany(4, 2),
        ManyToMany(5, 2),
    ]

students_dict = {student.student_id: student for student in students}
groups_dict = {group.group_id: group for group in groups}

# Запрос 1
print("Задание 1")
for group in groups:
    if group.name.startswith('A'):
        print(group.name)
        for student in group.students:
            print(f"\t{student.name} {student.birth_year}")

# Запрос 2
print("\nЗадание 2")
max_birth_years = {}
for group in groups:
    max_year = max(student.birth_year for student in group.students)
    max_birth_years[group.name] = max_year

for group, year in max_birth_years.items():
    print((group, year))

# Запрос 3
print("\nЗадание 3")
for group in groups:
    print(group.name)
    for student in group.students:
        print(f"\t{student.name}")

```

## test.py:

```

# test.py
import unittest
from main import Group, Student, ManyToMany

class TestGroup(unittest.TestCase):
    def test_add_student(self):
        group = Group(1, "TestGroup")
        student = Student(1, "TestStudent", 2000, group)

        group.add_student(student)
        self.assertIn(student, group.students)

class TestStudent(unittest.TestCase):
    def test_student_str(self):
        group = Group(1, "TestGroup")
        student = Student(1, "TestStudent", 2000, group)
        self.assertEqual(str(student), "TestStudent (2000)")

class TestManyToMany(unittest.TestCase):
    def test_many_to_many_creation(self):
        many_to_many = ManyToMany(1, 2)
        self.assertEqual(many_to_many.student_id, 1)

```

```
self.assertEqual(many_to_many.group_id, 2)

if name == 'main':
    unittest.main()
```

## Результаты Тестов

Testing started at 03:38 ...

Launching unittests with arguments python -m unittest /Users/razzle/PycharmProjects/RK

Ran 3 tests in 0.001s

OK