

ИУ5-33Б

РК2 по ПиКЯП

Вариант:

Вариант В.

1. «Отдел» и «Сотрудник» связаны соотношением один-ко-многим.
Выведите список всех сотрудников, у которых фамилия начинается с буквы «А», и названия их отделов.
2. «Отдел» и «Сотрудник» связаны соотношением один-ко-многим.
Выведите список отделов с минимальной зарплатой сотрудников в каждом отделе, отсортированный по минимальной зарплате.
3. «Отдел» и «Сотрудник» связаны соотношением многие-ко-многим.
Выведите список всех связанных сотрудников и отделов, отсортированный по сотрудникам, сортировка по отделам произвольная.

12	Язык программирования	Средство разработки
----	-----------------------	---------------------

Текст программы:

```
import unittest
```

```
class ProgrammingLanguage:
```

```
    """Язык программирования"""
```

```
    def __init__(self, id, name, ide_id):
```

```
        self.id = id
```

```
        self.name = name
```

```
        self.ide_id = ide_id
```

```
class IDE:
```

```
    """Средство разработки (IDE)"""
```

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

```
        self.id = id
```

```
        self.name = name
```

```
class LanguageIDE:
```

```
    """
```

```
    'Языки программирования в средах разработки' для реализации
```

```
    связи многие-ко-многим
```

```
    """
```

```
    def __init__(self, ide_id, language_id, users_count):
```

```
        self.ide_id = ide_id
```

```
        self.language_id = language_id
```

```
        self.users_count = users_count
```

```
# Список средств разработки (IDE)
```

```
ides = [
```

```
    IDE(1, 'PyCharm'),
```

```
    IDE(2, 'Visual Studio Code'),
```

```
    IDE(3, 'IntelliJ IDEA'),
```

```
    IDE(4, 'Eclipse'),
```

```
    IDE(5, 'Sublime Text'),
```

```
]
```

```
# Список языков программирования
```

```
languages = [  
    ProgrammingLanguage(1, 'Python', 1),  
    ProgrammingLanguage(2, 'Java', 3),  
    ProgrammingLanguage(3, 'C++', 4),  
    ProgrammingLanguage(4, 'JavaScript', 2),  
    ProgrammingLanguage(5, 'C#', 3),  
]
```

```
# Связь языков программирования и средств разработки
```

```
languages_ides = [  
    LanguageIDE(1, 1, 1000),  
    LanguageIDE(2, 4, 2000),  
    LanguageIDE(3, 2, 1500),  
    LanguageIDE(4, 3, 1800),  
    LanguageIDE(5, 5, 800),  
    LanguageIDE(1, 4, 500),  
    LanguageIDE(3, 5, 300),  
    LanguageIDE(2, 2, 400),  
]
```

```
class TestProgrammingLanguage(unittest.TestCase):
```

```
    def test_programming_language_init(self):  
        lang = ProgrammingLanguage(1, "Python", 1)  
        self.assertEqual(lang.id, 1)  
        self.assertEqual(lang.name, "Python")  
        self.assertEqual(lang.ide_id, 1)
```

```
class TestIDE(unittest.TestCase):  
    def test_ide_init(self):  
        ide = IDE(1, "PyCharm")  
        self.assertEqual(ide.id, 1)  
        self.assertEqual(ide.name, "PyCharm")  
  
class TestLanguageIDE(unittest.TestCase):  
    def test_language_ide_init(self):  
        lang_ide = LanguageIDE(1, 1, 1000)  
        self.assertEqual(lang_ide.ide_id, 1)  
        self.assertEqual(lang_ide.language_id, 1)  
        self.assertEqual(lang_ide.users_count, 1000)  
  
if __name__ == '__main__':  
    unittest.main()
```

```
Ran 3 tests in 0.020s
```

```
OK
```