**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»

Отчет по РК №1

Вариант запросов: Г

Вариант предметной области: 12

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-34Б |  | преподаватель каф. ИУ5 |
| Петров Никита |  | Гапанюк Ю. Е. |
|  |  |  |

Москва, 2023 г.

**Вариант запросов Г. Предметная область 12.**

# «Язык программирования» и «Средство разработки» связаны соотношением один-ко-многим. Выведите список всех языков, начинающихся с буквы ‘P’.

# «Язык программирования» и «Средство разработки» связаны соотношением один-ко-многим. Выведите список средств разработки с максимальным годом создания поддерживаемого языка, отсортированный по максимальному году создания.

# «Язык программирования» и «Средства разработки» связаны соотношением многие-ко-многим. Выведите список всех связанных языков программирования и средств разработки, отсортированный по средам разработки, сортировка по языкам программирования произвольная.

# **Задание**

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).

**Листинг программы.**

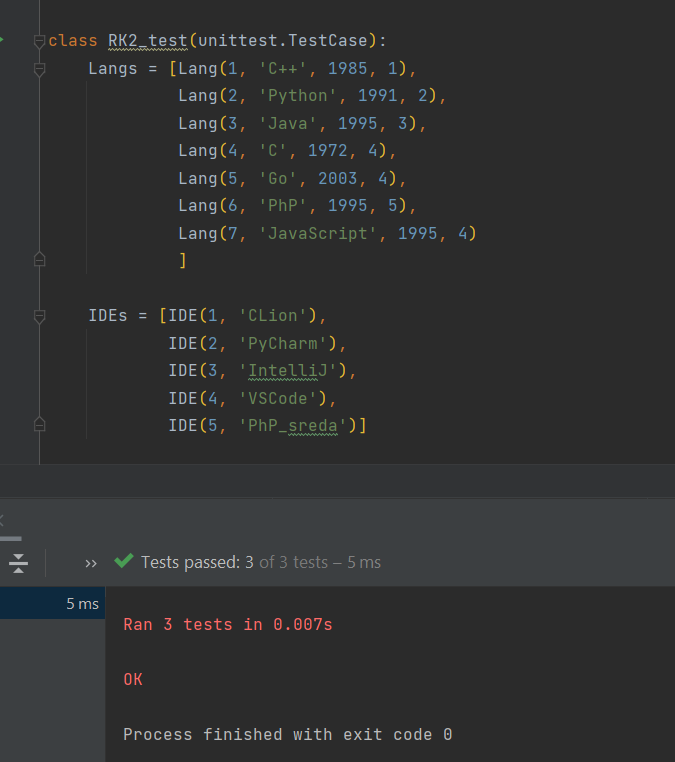
**Файл main.py**

from operator import itemgetter  
  
  
class Lang:  
 def \_\_init\_\_*(*self, id, name, year, ide\_id*)*:  
 self.id = id  
 self.name = name  
 self.year = year  
 self.ide\_id = ide\_id  
  
 def \_\_str\_\_*(*self*)*:  
 return f"*{*self.name*}*"  
  
  
class IDE:  
 def \_\_init\_\_*(*self, id, name*)*:  
 self.id = id  
 self.name = name  
  
 def \_\_str\_\_*(*self*)*:  
 return f"*{*self.name*}*"  
  
  
class LangIDE:  
 def \_\_init\_\_*(*self,lang\_id, ide\_id*)*:  
 self.lang\_id = lang\_id  
 self.ide\_id = ide\_id  
  
 def \_\_str\_\_*(*self*)*:  
 return f"*{*self.ide\_id*}* - *{*self.lang\_id*}*"  
  
  
Langs = *[*Lang*(*1, 'C++', 1985, 1*)*,  
 Lang*(*2, 'Python', 1991, 2*)*,  
 Lang*(*3, 'Java', 1995, 3*)*,  
 Lang*(*4, 'C', 1972, 4*)*,  
 Lang*(*5, 'Go', 2003, 4*)*,  
 Lang*(*6, 'PhP', 1995, 5*)*,  
 Lang*(*7, 'JavaScript', 1995,4*)  
 ]*IDEs = *[*IDE*(*1, 'CLion'*)*,  
 IDE*(*2, 'PyCharm'*)*,  
 IDE*(*3, 'IntelliJ'*)*,  
 IDE*(*4, 'VSCode'*)*,  
 IDE*(*5, 'PhP\_sreda'*)]*Lang\_IDE = *[*LangIDE*(*1, 1*)*,  
 LangIDE*(*2, 2*)*,  
 LangIDE*(*3, 3*)*,  
 LangIDE*(*1, 4*)*,  
 LangIDE*(*4, 4*)*,  
 LangIDE*(*5, 4*)*,  
 LangIDE*(*6, 5*)*,  
 LangIDE*(*7, 4*)]*def main*()*:  
 one\_to\_many = *[(*l.name, l.year, i.name*)* for l in Langs  
 for i in IDEs  
 if l.ide\_id == i.id*]* many\_to\_many\_temp = *[(*i.name, li.ide\_id, li.lang\_id*)* for i in IDEs  
 for li in Lang\_IDE  
 if i.id == li.ide\_id  
 *]* many\_to\_many = *[(*l.name, l.year, ide\_name*)* for ide\_name, ide\_id, lang\_id in many\_to\_many\_temp  
 for l in Langs if l.id == lang\_id*]* print*(*"Задание 1"*)* ans1 = *[]* for i in one\_to\_many:  
 if i*[*0*][*0*]* == "P":  
 ans1.append*(*i*[*0*])* print*(*ans1*)* print*(*"Задание 2"*)* ans2\_uns = *[]* for instr in IDEs:  
 instr\_langs = list*(*filter*(*lambda i: i*[*2*]* == instr.name, one\_to\_many*))* if len*(*instr\_langs*)* > 0:  
 years = *[*year for \_, year, \_ in instr\_langs*]* max\_ = max*(*years*)* ans2\_uns.append*((*instr.name, max\_*))* ans2 = sorted*(*ans2\_uns, key=itemgetter*(*1*))* print*(*ans2*)* print*(*'Задание 3'*)* print*(*sorted*(*many\_to\_many, key=lambda item: *(*item*[*2*]*, item*[*1*])))*if \_\_name\_\_ == '\_\_main\_\_':  
 main*()*

**Файл tests.py**

import unittest  
from main import \*  
  
  
class Rk2\_test*(*unittest.TestCase*)*:  
 Langs = *[*Lang*(*1, 'C++', 1985, 1*)*,  
 Lang*(*2, 'Python', 1991, 2*)*,  
 Lang*(*3, 'Java', 1995, 3*)*,  
 Lang*(*4, 'C', 1972, 4*)*,  
 Lang*(*5, 'Go', 2003, 4*)*,  
 Lang*(*6, 'PhP', 1995, 5*)*,  
 Lang*(*7, 'JavaScript', 1995, 4*)  
 ]* IDEs = *[*IDE*(*1, 'CLion'*)*,  
 IDE*(*2, 'PyCharm'*)*,  
 IDE*(*3, 'IntelliJ'*)*,  
 IDE*(*4, 'VSCode'*)*,  
 IDE*(*5, 'PhP\_sreda'*)]* Lang\_IDE = *[*LangIDE*(*1, 1*)*,  
 LangIDE*(*2, 2*)*,  
 LangIDE*(*3, 3*)*,  
 LangIDE*(*1, 4*)*,  
 LangIDE*(*4, 4*)*,  
 LangIDE*(*5, 4*)*,  
 LangIDE*(*6, 5*)*,  
 LangIDE*(*7, 4*)]* def test\_1*(*self*)*:  
 one\_to\_many = *[(*l.name, l.year, i.name*)* for l in Langs  
 for i in IDEs  
 if l.ide\_id == i.id*]* ans1 = *[]* for i in one\_to\_many:  
 if i*[*0*][*0*]* == "P":  
 ans1.append*(*i*[*0*])* self.assertEqual*(*ans1, *[*'Python', 'PhP'*])* def test\_2*(*self*)*:  
 one\_to\_many = *[(*l.name, l.year, i.name*)* for l in Langs  
 for i in IDEs  
 if l.ide\_id == i.id*]* ans2\_uns = *[]* for instr in IDEs:  
 instr\_langs = list*(*filter*(*lambda i: i*[*2*]* == instr.name, one\_to\_many*))* if len*(*instr\_langs*)* > 0:  
 years = *[*year for \_, year, \_ in instr\_langs*]* max\_ = max*(*years*)* ans2\_uns.append*((*instr.name, max\_*))* ans2 = sorted*(*ans2\_uns, key=itemgetter*(*1*))* self.assertEqual*(*ans2, *[(*'CLion', 1985*)*, *(*'PyCharm', 1991*)*, *(*'IntelliJ', 1995*)*,  
 *(*'PhP\_sreda', 1995*)*, *(*'VSCode', 2003*)])* def test\_3*(*self*)*:  
 many\_to\_many\_temp = *[(*i.name, li.ide\_id, li.lang\_id*)* for i in IDEs  
 for li in Lang\_IDE  
 if i.id == li.ide\_id  
 *]* many\_to\_many = *[(*l.name, l.year, ide\_name*)* for ide\_name, ide\_id, lang\_id in many\_to\_many\_temp  
 for l in Langs if l.id == lang\_id*]* ans3 = sorted*(*many\_to\_many, key=lambda item: *(*item*[*2*]*, item*[*1*]))* self.assertEqual*(*ans3, *[(*'C++', 1985, 'CLion'*)*, *(*'Java', 1995, 'IntelliJ'*)*, *(*'PhP', 1995, 'PhP\_sreda'*)*,  
 *(*'Python', 1991, 'PyCharm'*)*, *(*'C', 1972, 'VSCode'*)*, *(*'C++', 1985, 'VSCode'*)*,  
 *(*'JavaScript', 1995, 'VSCode'*)*, *(*'Go', 2003, 'VSCode'*)])*if \_\_name\_\_ == '\_\_main\_\_':  
 unittest.main*()*

**Результат выполнения программы**

****