|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н. Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н. Э. Баумана)** |

ФАКУЛЬТЕТ «Информатика и системы управления»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Отчет**

**по лабораторной работе № 5**

|  |  |
| --- | --- |
| **Дисциплина:** Функциональное и логическое программирование  **Студент:** Платонова Ольга  **Группа:** ИУ7-65Б  **Преподаватели:** Толпинская Н. Б.  Строганов Ю. В. |  |

Москва, 2021 г.

***Задание***

Создать базу знаний «Собственники», дополнив базу знаний, хранящую знания:

* «Телефонный справочник»: фамилия, №тел, адрес – структура (город, улица, №дома, №кв)
* «Автомобили»: Фамилия\_владельца, марка, цвет, стоимость и др.
* «Вкладчики банков»: фамилия, банк, счет, сумма, др.

знаниями о дополнительной собственности владельца. Преобразовать знания об автомобиле к форме знаний о собственности.

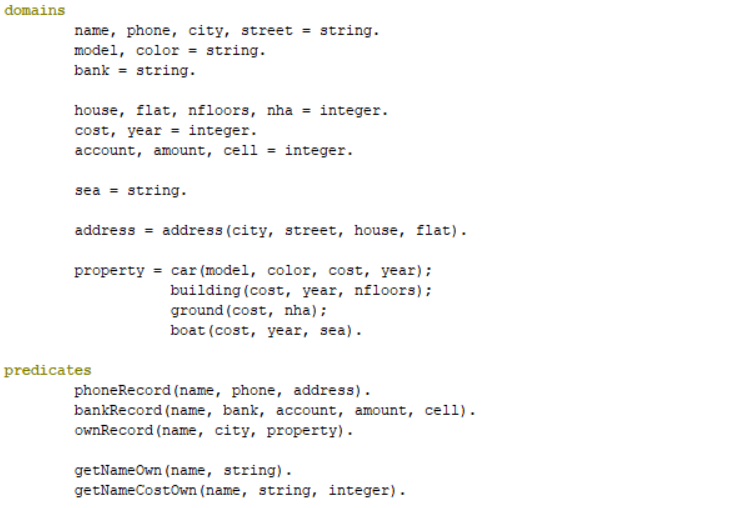
Вид собственности:

* Строение, стоимость и другие его характеристики;
* Участок, стоимость и другие его характеристики;
* Водный\_транспорт, стоимость и другие его характеристики.

Обеспечить возможность поиска:

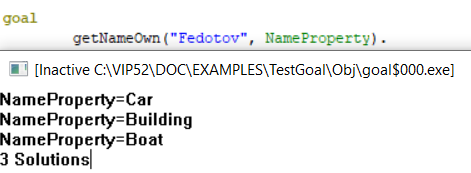
1. Названий всех объектов собственности заданного субъекта,
2. Названий и стоимости всех объектов собственности заданного субъекта.

***Листинг***

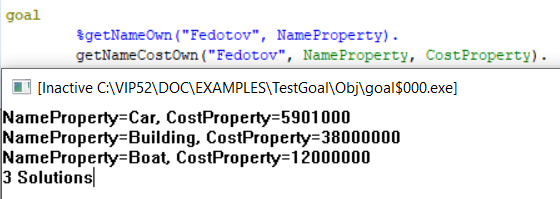




1. Названия всех объектов собственности заданного субъекта.



1. Названия и стоимости всех объектов собственности заданного субъекта.



**getNameCostOwn("Fedotov", NameProperty, CostProperty).**

|  |  |  |
| --- | --- | --- |
| № шага | Сравниваемые термы; результат; подстановка | Дальнейшие действия |
| 1 | getNameCostOwn("Fedotov", NameProperty, CostProperty).  phoneRecord(…)  Неудача. Не унифицируемые. | Переход к следующему заголовку БЗ. |
|  | … |  |
| 2 | getNameCostOwn("Fedotov", NameProperty, CostProperty).  getNameCostOwn(Name, NProp, CProp).  Успех. Унифицируемые.  Подстановка: {Name = “Fedotov”, NProp = NameProperty, CProp = CProperty}. | Удаляется из стека:  getNameCostOwn("Fedotov", NameProperty, CostProperty).  Новая резольвента:  ownRecord(“Fedotov”, \_, car(\_, \_, CProp, \_)),  NProp = "Car". |
| 3 | ownRecord(“Fedotov”, \_, car(\_, \_, CProp, \_)).  phoneRecord(…).  Неудача. Не унифицируемые. | Переход к следующему заголовку БЗ. |
|  | … |  |
| 4 | ownRecord(“Fedotov”, \_, car(\_, \_, CProp, \_)).  ownRecord("Fedotov", "Piter", car("Lambargini", "red", 5901000, 2021)).  Успех. Унифицируемые.  Подстановка: {“Fedotov” = ”Fedotov”, “Piter” = \_, “Lambargini” = \_, ”red” = \_, 5901000 = CProp, 2021 = \_}. | Удаляется из стека:  ownRecord(“Fedotov”, \_, car(\_, \_, CProp, \_)).  Новая резольвента:  NProp = "Car". |
| 5 |  | Удаляется из стека:  NProp = "Car".  Выводится “Fedotov”, “Car”, 5901000  Новая резольвента:  ownRecord(“Fedotov”, \_, car(\_, \_, CProp, \_)),  NProp = "Car". |
| 6 | ownRecord(“Fedotov”, \_, car(\_, \_, CProp, \_)).  ownRecord("Fedotov", "Piter", building(38000000, 2020, 5)).  Неудача. Не унифицируемые. | Переход к следующему заголовку БЗ. |
|  | … |  |
| 7 | ownRecord(“Fedotov”, \_, car(\_, \_, CProp, \_)).  getNameCostOwn().  Неудача. Не унифицируемые. | Все факты пройдены.  Удаляется из стека:  ownRecord(“Fedotov”, \_, car(\_, \_, CProp, \_)),  NProp = "Car".  Новая резольвента:  getNameCostOwn("Fedotov", NameProperty, CostProperty). |
| 8 | getNameCostOwn("Fedotov", NameProperty, CostProperty).  getNameCostOwn(Name, NProp, CProp).  Успех. Унифицируемые.  Подстановка: {Name = “Fedotov”, NProp = NameProperty, CProp = CProperty}. | Удаляется из стека:  getNameCostOwn("Fedotov", NameProperty, CostProperty).  Новая резольвента:  ownRecord("Fedotov", \_, building(Cprop, \_, \_)),  NProp = "Building". |
| 9 | ownRecord("Fedotov", \_, building(CProp, \_, \_)).  phoneRecord(…)  Неудача. Не унифицируемые. | Переход к следующему заголовку БЗ. |
|  | … |  |
| 10 | ownRecord("Fedotov", \_, building(CProp, \_, \_)).  ownRecord("Fedotov", "Piter", building(38000000, 2020, 5)).  Успех. Унифицируемые.  Подстановка: {“Fedotov” = ”Fedotov”, “Piter” = \_, 38000000 = CProp, 2020 = \_, 5 = \_}. | Удаляется из стека:  ownRecord("Fedotov", \_, building(CProp, \_, \_)).  Новая резольвента:  NProp = "Building". |
| 11 |  | Удаляется из стека:  NProp = "Building".  Выводится “Fedotov”, “Building”, 38000000  Новая резольвента:  ownRecord("Fedotov", \_, building(CProp, \_, \_)),  NProp = "Building". |
| 12 | ownRecord("Fedotov", \_, building(CProp, \_, \_)).  ownRecord("Fedotov", "Piter", boat(12000000, 2019, "Red")).  Неудача. Не унифицируемые. | Переход к следующему заголовку БЗ. |
| 13 | … |  |
| 14 | ownRecord("Fedotov", \_, building(CProp, \_, \_)).  getNameCostOwn().  Неудача. Не унифицируемые. | Все факты пройдены.  Удаляется из стека:  ownRecord("Fedotov", \_, building(CProp, \_, \_)),  NProp = "Building".  Новая резольвента:  getNameCostOwn("Fedotov", NameProperty, CostProperty). |
| 15 | getNameCostOwn("Fedotov", NameProperty, CostProperty).  getNameCostOwn(Name, NProp, CProp).  Успех. Унифицируемые.  Подстановка: {Name = “Fedotov”, NProp = NameProperty, CProp = CProperty}. | Удаляется из стека:  getNameCostOwn("Fedotov", NameProperty, CostProperty).  Новая резольвента:  ownRecord("Fedotov", \_, ground(CProp, \_)),  NProp = "Ground". |
| 16 | ownRecord(Name, \_, ground(CProp, \_)).  phoneRecord(…)  Неудача. Не унифицируемые. | Переход к следующему заголовку БЗ. |
|  | … |  |
| 17 | ownRecord(Name, \_, ground(CProp, \_)).  getNameCostOwn().  Неудача. Не унифицируемые. | Все факты пройдены.  Удаляется из стека:  ownRecord("Fedotov", \_, ground(CProp, \_)),  NProp = "Ground".  Новая резольвента:  getNameCostOwn("Fedotov", NameProperty, CostProperty). |
| 18 | getNameCostOwn("Fedotov", NameProperty, CostProperty).  getNameCostOwn(Name, NProp, CProp).  Успех. Унифицируемые.  Подстановка: {Name = “Fedotov”, NProp = NameProperty, CProp = CProperty}. | Удаляется из стека:  getNameCostOwn("Fedotov", NameProperty, CostProperty).  Новая резольвента:  ownRecord("Fedotov", \_, boat(CProp, \_, \_)),  NProp = "Boat". |
| 19 | ownRecord("Fedotov", \_, boat(CProp, \_, \_)).  phoneRecord(…)  Неудача. Не унифицируемые. | Переход к следующему заголовку БЗ. |
|  | … |  |
| 20 | ownRecord("Fedotov", \_, boat(CProp, \_, \_)).  ownRecord("Fedotov", "Piter", boat(12000000, 2019, "Red")).  Успех. Унифицируемые.  Подстановка: {“Fedotov” = ”Fedotov”, “Piter” = \_, 12000000= CProp, 2019 = \_, “Red” = \_}. | Удаляется из стека:  ownRecord("Fedotov", \_, boat(CProp, \_, \_)).  Новая резольвента:  NProp = "Boat". |
| 21 |  | Удаляется из стека:  NProp = "Boat".  Выводится “Fedotov”, “Boat”, 12000000  Новая резольвента:  ownRecord("Fedotov", \_, boat(CProp, \_, \_)),  NProp = "Boat". |
| 22 | ownRecord("Fedotov", \_, boat(CProp, \_, \_)).  getNameOwn().  Неудача. Не унифицируемые. | Переход к следующему заголовку БЗ. |
| 23 | … |  |
| 24 | ownRecord("Fedotov", \_, boat(CProp, \_, \_)).  getNameCostOwn().  Неудача. Не унифицируемые. | Все факты пройдены.  Удаляется из стека:  ownRecord("Fedotov", \_, boat(CProp, \_, \_)),  NProp = "Boat".  Новая резольвента:  getNameCostOwn("Fedotov", NameProperty, CostProperty). |
| 25 | getNameCostOwn("Fedotov", NameProperty, CostProperty). | Все факты пройдены.  Удаляется из стека:  getNameCostOwn("Fedotov", NameProperty, CostProperty).  Резольвента пуста. |

***Вывод***

1. *В каком фрагменте программы сформулировано знание? Это знание о чем на формальном уровне?*

Знание содержится в заголовке предложений базы знаний. А предложения - в разделе CLAUSES. Знание о том, что между аргументами терма (тела правила) существует отношение.

1. *Что содержит тело правила?*

Тело содержит условие истинности знания.

1. *Что дает использование переменных при формулировании знаний? В чем отличие формулировки знания с помощью термов с одинаковой арностью при использовании одной переменной и при использовании нескольких переменных?*

Переменная – способ обобщить. Предназначена для передачи значений «во времени и в пространстве». Чем больше переменных, тем более общий результат.

1. *С каким квантором переменные входят в правило, в каких пределах переменная уникальна?*

Переменные в факты и правила входят с квантором всеобщности («любой» элемент из множества). Именованные переменные уникальных в рамках предложения, а анонимная – любая уникальная.

1. *Какова семантика предложений раздела DOMAINS? Когда, где и с какой целью используется это описание?*

DOMAINS – раздел описания доменов. В разделе объявляются любые нестандартные домены в формате: <имя домена> = <определение домена>. Используется с целью описания имен и семантики доменов, когда природа или структура объектов не может быть определена с помощью стандартных доменов.

1. *Какова семантика предложений раздела PREDICATES? Когда, где и с какой целью используется это описание?*

PREDICATES – раздел описания предикатов. Раздел содержит описание структуры предиката. Используется с целью указания доменов, когда природа или структура объектов важна во время работы системы.

1. *Унификация каких термов запускается на самом первом шаге работы системы? Каковы назначение и результат использования алгоритма унификации?*

Вопроса и заголовка первого предложения базы знаний.

Алгоритм унификации необходим для того, чтобы подобрать знание, чтобы ответить на поставленный вопрос. Результатом работы алгоритма является значение переменной «неудача». Если неудача = 1, то унификация невозможна; если неудача = 0, то побочным действием работы алгоритма является содержимое результирующей ячейки – результирующая подстановка.

1. *В каком случае запускается механизм отката?*

В случае, когда унификация на текущем шаге завершается тупиковой ситуацией, или был получен ответ «да».