|  |  |
| --- | --- |
| 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» (ИУ7)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.03.04 Программная инженерия**

**Отчет**

|  |  |
| --- | --- |
| **по лабораторной работе №** | 15 |
|  |  |

**Дисциплина:** Функциональное и логическое программирование

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ7-62Б |  |  | Е.В. Брянская |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  | Н.Б.Толпинская  Ю.В.Строганов |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

Москва, 2021

**Задание**

Создать базу знаний «Собственники», переформатировать имеющуюся, используя вариантный домен.

* «Телефонный справочник»: Фамилия, №телефона, Адрес – структура (Город, Улица, №дома, №квартиры)
* «Автомобили»: Фамилия\_владельца, Марка, Цвет, Стоимость
* «Вкладчики банков»: Фамилия, Банк, счет, сумма

Вид собственности:

* Строение, стоимость и др.
* Участок, стоимость и др.
* Водный\_транспорт, стоимость и др.

Владелец может иметь только один объект каждого вида собственности, или не иметь некоторых видов собственности.

Используя конъюнктивное правило и разные формы задания одного вопроса (пояснять для какого № задания – какой вопрос), обеспечить возможность поиска:

1. Названий всех объектов собственности заданного субъекта
2. Названий и стоимости всех объектов собственности заданного субъекта
3. \* Разработать правило, позволяющее найти суммарную стоимость всех объектов собственности заданного субъекта

Для 2ого пункта и одной фамилии составить таблицу, отражающую конкретный порядок работы системы, с объяснениями порядка работы и особенностей использования доменов (указать конкретные Т1, Т2 и полную подстановку на каждом шаге)

При желании можно усложнить свою БЗ, введя варианты:

Строение: (Дом, офис, торговый центр)

Участок: (садовый, территория под застройку, территория под агро-работы)

Водный транспорт: (варианты названий)

|  |
| --- |
| DOMAINS  surname, phone = symbol.    city, street = symbol.  home, flat = integer.    brand, color = symbol.  price = real.  years = integer.    bank, account = symbol.  sum = real.    address = address(city, street, home, flat).    s = real.  property = building(price, s);  area(price, s);  water\_transport(price, years);  phone\_book(phone, address);  car(brand, color, price, years).    cur\_property = symbol.    PREDICATES  investor(surname, bank, account, sum).  phone\_book(surname, phone, address).    own(surname, property).    property\_price\_by\_owner(surname, cur\_property, price).    get\_price\_by\_owner(surname, cur\_property, price).    sum\_price(Surname, sum)  CLAUSES  investor("Birukova", "New\_1", deposit, 1000000).  investor("Mironova", "Old\_bank", deposit, 5000).  investor("Nikiforov", "VTB", deposit, 5000).    phone\_book("Birukova", "+123456", address("Moscow", "Zhukovsky Street", 12, 145)).  phone\_book("Mironova", "+333333", address("Ramenskoe", "New Street", 77, 77)).    own("Birukova", building(1000000, 50)).  own("Birukova", area(750000, 120)).  own("Birukova", water\_transport(5000000, 1)).  own("Filin", water\_transport(55000000, 2)).    own("Birukova", car("BMW", black, 4500000, 2)).  own("Filin", car("Honda", grey, 2300000, 4)).  own("Nikiforov", car("Honda", grey, 2000000, 6)).    property\_price\_by\_owner(Surname, building, Price) :- own(Surname, building(Price, \_)).  property\_price\_by\_owner(Surname, area, Price) :- own(Surname, area(Price, \_)).  property\_price\_by\_owner(Surname, water\_transport, Price) :- own(Surname, water\_transport(Price, \_)).  property\_price\_by\_owner(Surname, car, Price) :- own(Surname, car(\_, \_, Price, \_)).    get\_price\_by\_owner(Surname, building, Price) :- own(Surname, building(Price, \_)), !.  get\_price\_by\_owner(Surname, area, Price) :- own(Surname, area(Price, \_)), !.  get\_price\_by\_owner(Surname, water\_transport, Price) :- own(Surname, water\_transport(Price, \_)), !.  get\_price\_by\_owner(Surname, car, Price) :- own(Surname, car(\_, \_, Price, \_)), !.  get\_price\_by\_owner(\_, \_, 0).    sum\_price(Surname, Sum) :-  get\_price\_by\_owner(Surname, building, Price1),  get\_price\_by\_owner(Surname, area, Price2),  get\_price\_by\_owner(Surname, water\_transport, Price3),  get\_price\_by\_owner(Surname, car, Price4),  Sum = Price1 + Price2 + Price3 + Price4.    GOAL  %property\_price\_by\_owner("Filin", Property, \_).  %property\_price\_by\_owner("Filin", Property, Price).  %sum\_price("Nikiforov", Sum). |

1. Найти названия всех объектов собственности заданного субъекта

|  |  |
| --- | --- |
| property\_price\_by\_owner("Birukova ", Property, \_). |  |
| property\_price\_by\_owner("Filin", Property, \_). |  |
| property\_price\_by\_owner("Mishina", Property, \_). |  |

1. Найти названия и стоимости всех объектов собственности заданного субъекта

|  |  |
| --- | --- |
| property\_price\_by\_owner("Birukova", Property, Price). |  |
| property\_price\_by\_owner("Filin", Property, Price). |  |
| property\_price\_by\_owner("Mishina", Property, Price). |  |

1. Разработать правило, позволяющее найти суммарную стоимость всех объектов собственности заданного субъекта

|  |  |
| --- | --- |
| sum\_price("Birukova", Sum). |  |
| sum\_price("Filin", Sum). |  |
| sum\_price("Mishina", Sum). |  |

Для 2ого пункта и одной фамилии составить таблицу, отражающую конкретный порядок работы системы, с объяснениями порядка работы и особенностей использования доменов (указать конкретные Т1, Т2 и полную подстановку на каждом шаге)

|  |  |  |
| --- | --- | --- |
| **№** | **Сравниваемые термы; результат; подстановка, если есть** | **Дальнейшие действие: прямой ход или откат (к чему приводит?)** |
| 0. |  | Начальное состояние резольвенты:  property\_price\_by\_owner("Filin", Propertyt, Pricet) |
|  | property\_price\_by\_owner("Filin", Propertyt, Pricet)  =  investor("Birukova", "New\_1", deposit, 1000000) | Неудача  (не совпали главные функторы) |
|  | \*\*\* | \*\*\* |
|  | property\_price\_by\_owner("Filin", Propertyt, Pricet)  =  phone\_book("Birukova", "+123456", address("Moscow", "Zhukovsky Street", 12, 145)) | Неудача  (не совпали главные функторы) |
|  | \*\*\* | \*\*\* |
|  | property\_price\_by\_owner("Filin", Propertyt, Pricet)  =  own("Birukova", building(1000000, 50)) | Неудача  (не совпали главные функторы) |
|  | \*\*\* | \*\*\* |
|  | property\_price\_by\_owner("Filin", Propertyt, Pricet)  =  property\_price\_by\_owner(Surname, building, Price)  **Постановка:**  {Surname=”Filin”, Propertyt=building, Price=Pricet} | Удача  **Изменение резольвенты:**  Новое состояние резольвенты:  own(”Filin”, building(Pricet, \_)) |
|  | own(”Filin”, building(Pricet, \_))  =  investor("Birukova", "New\_1", deposit, 1000000) | Неудача  (не совпали главные функторы) |
|  | \*\*\* | \*\*\* |
|  | own(”Filin”, building(Pricet, \_))  =  own("Birukova", building(1000000, 50)) | Неудача  (не совпали главные функторы) |
|  | \*\*\* | \*\*\* |
|  | own(”Filin”, building(Pricet, \_))  =  own("Filin", water\_transport(55000000, 2)) | Неудача  (не совпали функторы) |
|  | own(”Filin”, building(Pricet, \_))  =  own("Birukova", car("BMW", black, 4500000, 2)) | Неудача  (не совпали константы) |
|  | own(”Filin”, building(Pricet, \_))  =  own("Filin", car("Honda", grey, 2300000, 4)) | Неудача  (не совпали функторы) |
|  | \*\*\* | \*\*\* |
|  | own(”Filin”, building(Pricet, \_))  =  sum\_price(Surname, Sum) | Неудача  (не совпали функторы)  **Откат** (просмотрена вся БЗ) |
|  | **Подстановка:**  {} | **Изменение состояния резольвенты:**  property\_price\_by\_owner("Filin", Propertyt, Pricet) |
|  | property\_price\_by\_owner("Filin", Propertyt, Pricet)  =  property\_price\_by\_owner(Surname, area, Price)  **Постановка:**  {Surname=”Filin”, Propertyt=area, Price=Pricet} | Удача  **Изменение резольвенты:**  Новое состояние резольвенты:  own(”Filin”, area(Pricet, \_)) |
|  | own(”Filin”, area(Pricet, \_))  =  investor("Birukova", "New\_1", deposit, 1000000) | Неудача  (не совпали главные функторы) |
|  | \*\*\* | \*\*\* |
|  | own(”Filin”, area(Pricet, \_))  =  sum\_price(Surname, Sum) | Неудача  (не совпали главные функторы)  **Откат** (просмотрена вся БЗ) |
|  | **Подстановка:**  {} | **Изменение состояния резольвенты:**  property\_price\_by\_owner("Filin", Propertyt, Pricet) |
|  | property\_price\_by\_owner("Filin", Propertyt, Pricet)  =  property\_price\_by\_owner(Surname, water\_transport, Price)  **Постановка:**  {Surname=”Filin”, Propertyt= water\_transport, Price=Pricet} | Удача  **Изменение резольвенты:**  Новое состояние резольвенты:  own(”Filin”, water\_transport(Pricet, \_)) |
|  | own(”Filin”, water\_transport(Pricet, \_))  =  investor("Birukova", "New\_1", deposit, 1000000) | Неудача  (не совпали главные функторы) |
|  | \*\*\* | \*\*\* |
|  | own(”Filin”, water\_transport(Pricet, \_))  =  own("Filin", water\_transport(55000000, 2))  **Постановка:**  {Surname=”Filin”, Propertyt= water\_transport, Price=Pricet, Pricet=55000000} | Удача  **Изменение резольвенты:**  Новое состояние резольвенты:  Резольвента пуста  **Выводится:**  Propertyt= water\_transport,  Pricet=55000000  **Откат** |
|  | **Подстановка:**  {Surname=”Filin”, Propertyt= water\_transport, Price=Pricet} | **Изменение состояния резольвенты:**  own(”Filin”, water\_transport(Pricet, \_)) |
|  | own(”Filin”, water\_transport(Pricet, \_))  =  own("Birukova", car("BMW", black, 4500000, 2)) | Неудача  (не совпали константы) |
|  | \*\*\* | \*\*\* |
|  | own(”Filin”, water\_transport(Pricet, \_))  =  sum\_price(Surname, Sum) | Неудача  (не совпали главные функторы)  **Откат** (просмотрена вся БЗ) |
|  | **Подстановка:**  {} | **Изменение состояния резольвенты:**  property\_price\_by\_owner("Filin", Propertyt, Pricet) |
|  | property\_price\_by\_owner("Filin", Propertyt, Pricet)  =  property\_price\_by\_owner(Surname, car, Price)  **Постановка:**  {Surname=”Filin”, Propertyt= car, Price=Pricet} | Удача  **Изменение резольвенты:**  Новое состояние резольвенты:  own(”Filin”, car(\_, \_, Pricet, \_)) |
|  | own(”Filin”, car(\_, \_, Pricet, \_))  =  investor("Birukova", "New\_1", deposit, 1000000) | Неудача  (не совпали главные функторы) |
|  | \*\*\* | \*\*\* |
|  | own(”Filin”, car(\_, \_, Pricet, \_))  =  own("Filin", car("Honda", grey, 2300000, 4))  **Постановка:**  {Surname=”Filin”, Propertyt=car, Price=Pricet, Pricet=2300000} | Удача  **Изменение резольвенты:**  Новое состояние резольвенты:  Резольвента пуста  **Выводится:**  Propertyt=car,  Pricet=2300000  **Откат** |
|  | **Подстановка:**  {Surname=”Filin”, Propertyt=car, Price=Pricet} | **Изменение состояния резольвенты:**  own(”Filin”, car(Pricet, \_)) |
|  | own(”Filin”, car(Pricet, \_))  =  own("Nikiforov", car("Honda", grey, 2000000, 6)) | Неудача  (не совпали константы) |
|  | \*\*\* | \*\*\* |
|  | own(”Filin”, car(Pricet, \_))  =  sum\_price(Surname, Sum) | Неудача  (не совпали главные функторы)  **Откат** (просмотрена вся БЗ) |
|  | **Подстановка:**  {} | **Изменение состояния резольвенты:**  property\_price\_by\_owner("Filin", Propertyt, Pricet) |
|  | property\_price\_by\_owner("Filin", Propertyt, Pricet)  =  get\_price\_by\_owner(Surname, building, Price) | Неудача  (не совпали главные функторы) |
|  | \*\*\* | \*\*\* |
|  | property\_price\_by\_owner("Filin", Propertyt, Pricet)  =  sum\_price(Surname, Sum) | Неудача  (не совпали главные функторы)  Просмотрена вся БЗ  Завершение работы |

Вопросы:

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

Знание формулируется в разделе CLAUSES, это знания о предметной области.

1. Что содержит тело правила?

Тело правила содержит условие истинности.

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?

Переменные нужны для обобщения, в программе же необходимы для передачи значения в пространстве и во времени. Чем больше переменных, тем сильнее обобщённость, абстракция.

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

Переменные в правило входят с квантором всеобщности. Именованная переменная уникальна в рамках одного предложения, анонимная же уникальна всегда.

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

DOMAINS - раздел описания доменов, используется для определения структуры, позволяют описать природу аргументов.

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

PREDICATES - раздел описания предикатов, используется для отслеживания природы аргументов.

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

На самом первом шаге запускается унификация цели и первого терма. Унификация позволяет формализовать процесс логического вывода и обеспечивает двунаправленную передачу параметров процедурам, неразрушающее присваивание, проверку условий. Результат унификации – ответ да/нет.

1. В каком случае запускается механизм отката?

Механизм отката запускается, в случае если было найдено подходящие знание, позволяющее ответить «да» на поставленный вопрос (в таком случае система проверяет, нет ли еще решения), либо в случае, когда его, наоборот, найти не удалось, просмотрена вся база знаний.