|  |  |
| --- | --- |
| 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 ЛАБОРАТОРНЫМ РАБОТАМ №11,12,13**

Дисциплина Функциональное и логическое программирование

Тема «Структура программы на Prolog»

Студент Куликов Дмитрий Алексеевич

*фамилия, имя, отчество*

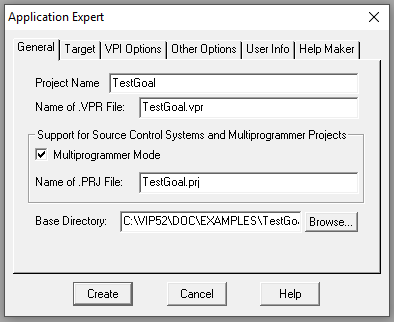
Группа ИУ7-62Б

Преподаватель **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  Толпинская Н.Б.

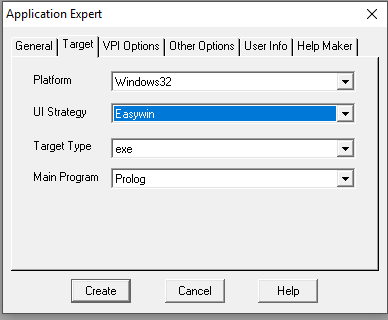
*подпись, дата фамилия, и.о.*

*2021 г*

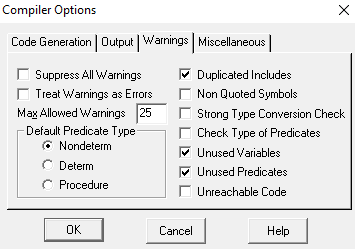
**Задание 1. Настроить TestGoal.**

****

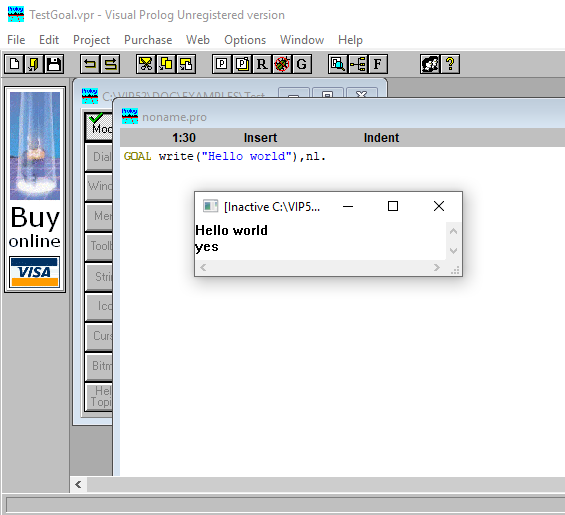
**Рис. 1. Создание нового проекта**

****

**Рис. 2. Настройка проекта на вкладке target**

****

**Рис. 3. Установки опций компилятора**

****

**Рис. 4. Запуск тестовой программы.**

**Задание 2. Составить программу – базу знаний, с помощью которой можно определить, например, множество студентов, обучающихся в одном ВУЗе. Студент может одновременно обучаться в нескольких ВУЗах. Привести примеры возможных вариантов вопросов и варианты ответов (не менее 3-х). Описать порядок формирования вариантов ответа.**

**Исходную базу знаний сформировать с помощью только фактов.**

**\*Исходную базу знаний сформировать, используя правила.**

**\*Разработать свою базу знаний (содержание произвольно).**

**Листинг 1.**

domains

name, surname, group, university = symbol.

age = integer.

predicates

study(name, surname, age, group, university).

studentByUniversity(name, surname, group, university).

clauses

study("Dmitry", "Kulikov", 20, "IU7-62B", "BMSTU").

study("Vladimir", "Markeev", 21, "112", "MRSU").

study("Dmitry", "Kulikov", 20, "112", "MRSU").

study("Danila", "Spirin", 21, "31", "FSIN").

study("Andrey", "Chalyy", 20, "32", "FSIN").

study("Nikita", "Garasev", 19, "IU7-62B", "BMSTU").

study("Andrey", "Chalyy", 20, "IU7-66B", "BMSTU").

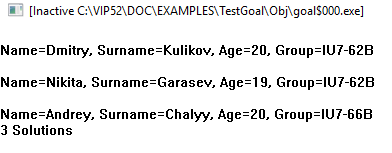
studentByUniversity(N, S, G, U) :- study(N, S, \_, G, U).

**1. Все студенты обучающиеся в BMSTU.**

goal

study(Name, Surname, Age, Group, "BMSTU"), nl.

Результат работы программы:

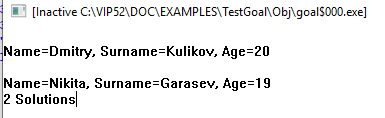
****

**2. Все студенты из группы IU7-62B и обучающиеся в BMSTU.**

goal

study(Name, Surname, Age, "IU7-62B", "BMSTU"), nl.

Результат работы программы:

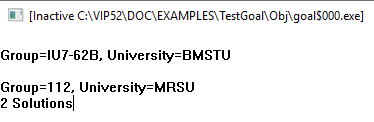
****

**3. Студент, обучающийся сразу в двух вузах.**

goal

study("Dmitry", "Kulikov", 20, Group, University), nl.

Результат работы программы:

****

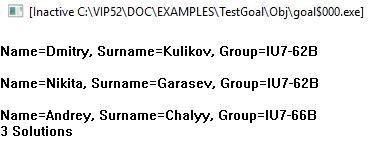
**4. Использование правила**

Получить всех студентов BMSTU.

goal

studentByUniversity(Name, Surname, Group, "BMSTU"), nl.

Результат работы программы:

****

**Задание 3.**

Составить программу, т.е. модель предметной области – базу знаний, объединив в ней информацию – 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.

Владелец может иметь несколько телефонов, автомобилей, вкладов (Факты).

Используя правила, обеспечить возможность поиска

1. а) По № телефона найти: Фамилию, Марку автомобиля, Стоимость автомобиля (может быть несколько),

в) Используя сформированное в пункте а) правило, по № телефона найти: только Марку автомобиля (автомобилей может быть несколько),

1. Используя простой, не составной вопрос: по Фамилии (уникальна в городе, но в разных городах есть однофамильцы) и Городу проживания найти: Улицу проживания, Банки, в которых есть вклады и №телефона.

Для задания 1 и задания 2:

для одного из вариантов ответов, и для а) и для в), описать словесно порядок поиска ответа на вопрос, указав, как выбираются знания, и, при этом, для каждого этапа унификации, выписать подстановку – наибольший общий унификатор, и соответствующие примеры термов.

domains

surname = symbol.

phone = symbol.

city = symbol.

street = symbol.

house = integer.

flat = integer.

adress = adress(city, street, house, flat).

brand = symbol.

color = symbol.

price = integer.

bank = symbol.

account = symbol.

amount = integer.

predicates

person(surname, phone, adress).

car(surname, brand, color, price).

depositor(surname, bank, account, amount).

carByPhone(phone, surname, brand, price).

modelByPhone(phone, brand).

accountAndPhoneBySurnameAndCity(surname, city, street, phone, bank).

clauses

person("Kulikov", "89271939173", adress("Saransk", "Kovalenko", 7, 89)).

person("Kulikov", "89272030284", adress("Saransk", "Kovalenko", 7, 89)).

person("Garasev", "89273141395", adress("Moscow", "Kapotnoua", 19, 110)).

person("Chalyy", "892722822832", adress("Anadyr", "Morshovaya", 18, 322)).

person("Kulikov", "89273222280", adress("Moscow", "Kurskya", 25, 2)).

person("Kulikov", "89273222281", adress("Moscow", "Kurskya", 25, 2)).

car("Kulikov", "Audi A6", "white", 4500000).

car("Kulikov", "Audi A4", "black", 1500000).

car("Garasev", "BMW X6", "pink", 7500000).

car("Chalyy", "Bugatti", "gold", 7500000).

car("Chalyy", "Lada Vesta", "white", 500000).

depositor("Kulikov", "Sberbank", "11111", 1500).

depositor("Kulikov", "Tinkoff", "22222", 3500).

depositor("Garasev", "Tinkoff", "22822", 7500).

depositor("Garasev", "VTB", "23822", 6500).

depositor("Chalyy", "SelhozBank", "33333", 500).

carByPhone(Phone, Surname, Brand, Price) :- person(Surname, Phone,\_),

car(Surname, Brand, \_, Price).

modelByPhone(Phone, Brand) :- carByPhone(Phone, \_, Brand, \_).

accountAndPhoneBySurnameAndCity(Surname, City, Street, Phone, Bank) :-

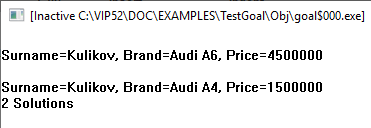
person(Surname, Phone, adress(City, Street, \_, \_)),

depositor(Surname, Bank, \_, \_).

**Результаты работы программы:**

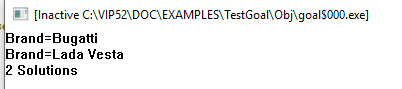
goal

carByPhone("89271939173", Surname, Brand, Price), nl.



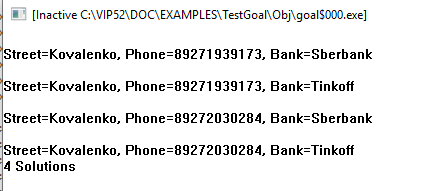
goal

modelByPhone("892722822832", Brand).



goal

accountAndPhoneBySurnameAndCity("Kulikov", "Saransk", Street, Phone, Bank), nl.



**1.**  carByPhone("89271939173", Surname, Brand, Price).

|  |  |  |
| --- | --- | --- |
| № шага | Сравниваемые термы; результат; подстановка, если есть | Дальнейшие действия |
| 0 | T1 = carByPhone(Phone, Surname, Brand, Price)  T2 = carByPhone("89271939173", Surname, Brand, Price)  Связываются через Phone = "89271939173" | carByPhone убирается из стека  person(Surname, Phone,\_)  заносится в стек  car(Surname, Brand, \_, Price)  заносится в стек |
| 1. | T1 = person(Surname, Phone,\_)  T2 = person("Kulikov", "89271939173", adress("Saransk", "Kovalenko", 7, 89)).  Связываются через Surname = "Kulikov" | person(Surname, Phone,\_)  убирается из стека |
| 2 | T1 = car("Kulikov", Brand, \_, Price).  T2 = car("Kulikov", "Audi A6", "white", 4500000).  Связываются через Brand = "Audi A6", Price = 4500000 | car(Surname, Brand, \_, Price)  убирается из стека  выводится “Kulikov”, “Audi A6” , 4500000  car(Surname, Brand, \_, Price)  заносится в стек  Brand и Price развязываются |
| 3 | T1 = car("Kulikov", Brand, \_, Price).  T2 = car("Kulikov", "Audi A4", "black", 1500000).  Связываются через Brand = "Audi A4", Price = 1500000 | car(Surname, Brand, \_, Price)  убирается из стека  выводится “Kulikov”, “Audi A4” , 1500000  car(Surname, Brand, \_, Price)  заносится в стек  Brand и Price развязываются |
| 4. | T1 = car("Kulikov", Brand, \_, Price)  T2 = car("Garasev", "BMW X6", "pink", 7500000).  Не унифицируются |  |
| 5. | T1 = car("Kulikov", Brand, \_, Price).  T2 = car("Chalyy", "Bugatti", "gold", 7500000).  Не унифицируются |  |
| 6 | T1 = car("Kulikov", Brand, \_, Price).  T2 = car("Chalyy", "Lada Vesta", "white", 500000).  Не унифицируются | Откат. Все факты car пройдены  В стек заносится последний вопрос person(Surname, Phone,\_)  Surname развязывается |
| 7 |  | Откат. Все факты person пройдены  В стек заносится последний вопрос carByPhone  Phone развязывается.  Система завершает работу |

**2.**  modelByPhone("892722822832", Brand).

|  |  |  |
| --- | --- | --- |
| № шага | Сравниваемые термы; результат; подстановка, если есть | Дальнейшие действия |
| 0 | T1 = modelByPhone(Phone, Brand).  T2 = modelByPhone("892722822832", Brand).  Связываются через Phone = "892722822832" | modelByPhone убирается из стека  carByPhone(Phone, \_, Brand, \_).  заносится в стек |
| 1. | T1 = carByPhone("892722822832", \_, Brand, \_).  T2 = carByPhone(phone, surname, brand, price).  Связываются через Phone = "892722822832" | carByPhone убирается из стека  person(Surname, Phone,\_)  заносится в стек  car(Surname, Brand, \_, Price)  заносится в стек |
| 2 | T1 = person(Surname, Phone,\_).  T2 = person("Chalyy", "892722822832", adress("Anadyr", "Morshovaya", 18, 322)).  Связываются через Surname = "Chalyy" | person(Surname, Phone,\_)  убирается из стека |
| 3 | T1 = car("Chalyy", Brand, \_, Price).  T2 = car("Chalyy", "Bugatti", "black", 1500000).  Связываются через Brand = "Bugatti" | car(Surname, Brand, \_, Price)  убирается из стека  выводится “Chalyy”, “Bugatti”  car(Surname, Brand, \_, Price)  заносится в стек  Brand развязывается |
| 4. | T1 = car("Chalyy", Brand, \_, Price).  T2 = car("Chalyy", "Lada Vesta", "white", 500000).  Связываются через Brand = " Lada Vesta" | car(Surname, Brand, \_, Price)  убирается из стека  выводится “Chalyy”, “Lada Vesta”  Brand развязывается |
| 5. |  | Откат. Все факты car пройдены  В стек заносится последний вопрос person(Surname, Phone,\_)  Surname развязывается |
| 6 |  | Откат. Все факты person пройдены  В стек заносится последний вопрос carByPhone  Phone развязывается.  Система завершает работу |

**3.**  accountAndPhoneBySurnameAndCity("Kulikov", "Saransk", Street, Phone, Bank).

|  |  |  |
| --- | --- | --- |
| № шага | Сравниваемые термы; результат; подстановка, если есть | Дальнейшие действия |
| 0 | T1 = accountAndPhoneBySurnameAndCity(Surname,City, Street, Phone, Bank).  T2 = accountAndPhoneBySurnameAndCity("Kulikov", "Saransk", Street, Phone, Bank).  Связываются через Surname = "Kulikov" , City = "Saransk" | accountAndPhoneBy  SurnameAndCity убирается из стека  person(Surname, Phone, adress(City, Street, \_, \_))  заносится в стек depositor(Surname, Bank, \_, \_).  заносится в стек |
| 1. | T1 = person("Kulikov", Phone, adress(City, ("Saransk", Street,\_,\_))  T2 = person("Kulikov", "89271939173", adress("Saransk", "Kovalenko", 7, 89)).  Связываются через Street = "Kovalenko" ,  Phone = "89271939173" | person убирается из стека |
| 2 | T1 = depositor("Kulikov", Bank, \_, \_).  T2 = depositor("Kulikov", "Sberbank", "11111", 1500).  Связываются через Bank = "Sberbank" | depositor(Surname, Bank, \_, \_).  убирается из стека  выводится “Kovalenko”, "89271939173", "Sberbank"  depositor(Surname, Bank, \_, \_).  заносится в стек  Bank развязывается |
| 3 | T1 = depositor("Kulikov", Bank, \_, \_).  T2 = depositor("Kulikov", "Tinkoff", "22222", 3500).  Связываются через Bank = "Tinkoff" | depositor(Surname, Bank, \_, \_).  убирается из стека  выводится “Kovalenko”, "89271939173", "Sberbank"  depositor(Surname, Bank, \_, \_).  заносится в стек  Bank развязывается  car(Surname, Brand, \_, Price)  заносится в стек  Brand развязывается |
| 4. | … (факты depositor не унифицируются) | Откат. Все факты depositor пройдены  В стек заносится последний вопрос person(Surname, Phone, adress(City, Street, \_, \_))  Phone Street развязывается |
| 5. |  | Откат. Все факты car пройдены  В стек заносится последний вопрос person(Surname, Phone,\_)  Surname развязывается |
| 6 | T1 = person("Kulikov", Phone, adress(City, ("Saransk", Street,\_,\_))  T2 = person("Kulikov", "89271939173", adress("Saransk", "Kovalenko", 7, 89)).  Связываются через Street = "Kovalenko" ,  Phone = " 89272030284" | person убирается из стека |
| 7 | T1 = depositor("Kulikov", Bank, \_, \_).  T2 = depositor("Kulikov", "Sberbank", "11111", 1500).  Связываются через Bank = "Sberbank" | depositor(Surname, Bank, \_, \_).  убирается из стека  выводится “Kovalenko”, "89271939173", "Sberbank"  depositor(Surname, Bank, \_, \_).  заносится в стек  Bank развязывается |
| 8 | T1 = depositor("Kulikov", Bank, \_, \_).  T2 = depositor("Kulikov", "Tinkoff", "22222", 3500).  Связываются через Bank = "Tinkoff" | depositor(Surname, Bank, \_, \_).  убирается из стека  выводится “Kovalenko”, "89271939173", "Sberbank"  depositor(Surname, Bank, \_, \_).  заносится в стек  Bank развязывается  car(Surname, Brand, \_, Price)  заносится в стек  Brand развязывается |
| 9 | … (факты depositor не унифицируются) | Откат. Все факты depositor пройдены  В стек заносится последний вопрос person(Surname, Phone, adress(City, Street, \_, \_))  Phone Street развязывается |
| 10 | … (факты person не унифицируются) | Откат. Все факты person пройдены  В стек заносится последний вопрос accountAndPhoneBy  SurnameAndCity  Surname и City развязывается/  Система завершает работу. |

**Контрольные вопросы:**

1. Что такое терм?

Терм — это объект знаний в Prolog. Термы могут быть константами, переменными или составными термами. В свою очередь константы могут быть числовыми, символьными атомами или строками. Переменные могут быть именованными или анонимными. Составные термы используются для обозначения отношений между объектами, они объединяют отдельные элементы знаний в единый объект.

2. Что такое предикат в матлогике (математике)?

Предикат в матлогике — это функция, которая возвращает одно из двух значений: 0 («ложь») или 1 («Истина»).

3. Что описывает предикат в Prolog?

В Prolog предикат описывает отношение между объектами. Это отношение также как в математике может быть истинным.

4. Назовите виды предложений в программе и приведите примеры таких предложений из Вашей программы. Какие предложения являются основными, а какие – не основными? Каковы: синтаксис и семантика (формальный смысл) этих предложений (основных и неосновных)?

Виды предложений в программе:

* Факты — с их помощью записываются знания.  
  Пример: car(petrov, volvo, 1223122, red).
* Правила — это обобщенная запись знаний и условий, при которых знание является истиной.   
  Пример: carByPhone(Phone,Surname,Brand,Price):- person(Surname,Phone, \_ ),

car(Surname, Brand, Price,\_).

* Вопросы — используются для выяснения истинности какого-либо знания и

Пример: carByPhone("89271939173”, Surname, Brand, Price).

Основные предложения — это предложения, которые не содержат переменных.

Неосновные предложения — это предложения, которые содержат переменные в момент фиксации программы.

Синтаксически предложения записываются как

A :- B1,... , Bn.

А — это заголовок правила, а B1,..., Bn — тело правила.

Правила состоят и из заголовка и из тела. Факты — частный случай правил, состоят только из заголовка. Вопросы состоят только из тела.

5. Каковы назначение, виды и особенности использования переменных в программе на Prolog? Какое предложение БЗ сформулировано в более общей – абстрактной форме: содержащее или не содержащее переменных?

Переменные предназначены для передачи знаний.

Переменные могут быть именованные и анонимные.

Если переменная не имеет значения, то она называется свободной, иначе — конкретизированной. Анонимные переменные не могут быть связаны со значением. Анонимные переменные используются в случаях, когда необходимо использовать переменную, однако ее значение не существенно.

Именованные переменные уникальны в рамках одного предложения, т. е. в разных предложениях одно и то же имя переменной может использоваться для обозначения разных объектов. Анонимные переменные уникальны везде.

Все переменные безтиповые, в процессе вычисления они могут связываться с любыми объектами.

Предложение содержащее переменные сформулировано в более общей-абстрактной форме, так как несколько знаний могут подойти под одно предложение.

6. Что такое подстановка?

Подстановкой называется множество пар, вида: { x i = t i } , где x i – переменная, а t i – терм. Каждой переменной xi в соответствие ставится терм ti.

7. Что такое пример терма? Как и когда строится? Как Вы думаете, система строит и хранит примеры?

Пример терма А — это терм B такой, что существует такая подстановка θ, что B = Aθ.

Примеры термов строятся в процессе унификации, когда происходит сравнение двух термов, с помощью подстановки всех возможных значений из базы знаний. Пример хранится в памяти для продолжения доказательства. Он уничтожается когда система заходит в тупик или на вопрос можно дать ответ «да».