|  |  |
| --- | --- |
| 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 №** | 11/12/13 |
|  |  |

**Дисциплина:** Функциональное и логическое программирование

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

Москва, 2021

**Л/р 11**

**Задание**

Составить программу, с помощью которой можно определить, например, множество студентов, обучающихся в одном вузе.

|  |
| --- |
| DOMAINS  name, surname, subj, unv = symbol.    PREDICATES  student(name, surname, unv).  teacher(name, surname, subj, unv).    CLAUSES  student("Ann", "Richy", "unv\_1").  student("Boby", "Adamson", "unv\_3").  student("Galya", "Backer", "unv\_2").  student("Gigi", "Dyson", "unv\_1").    student("Tanya", "Park", Unv):-student("Ann", "Richy", Unv).    teacher("Kristina", "Nikiforova", "physics", "unv\_2").  teacher("Mila", "Kolovanova", "english", "unv\_2").  teacher("Anton", "Popov", "drawing", "unv\_4").    GOAL  %student("Boby", "Adamson", "unv\_3").    %student("Galya", Surname, "unv\_2").    %teacher(\_, \_, \_, "unv\_2").    %teacher(Name, Surname, Subj, "unv\_2"). |

|  |  |  |
| --- | --- | --- |
| 1 | student("Boby", "Adamson", "unv\_3"). |  |
| 2 | student("Galya", Surname, "unv\_2"). |  |
| 3 | teacher(\_, \_, \_, "unv\_2"). |  |
| 4 | teacher(Name, Surname, Subj, "unv\_2"). |  |

**Л/р 12**

**Задание**

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.

|  |
| --- |
| DOMAINS  name, surname, subj, unv = symbol.    PREDICATES  student(name, surname, unv).  teacher(name, surname, subj, unv).    CLAUSES  student("Ann", "Richy", "unv\_1").  student("Boby", "Adamson", "unv\_3").  student("Boby", "Dicson", "unv\_1").  student("Galya", "Backer", "unv\_2").  student("Gigi", "Dyson", "unv\_1").  student("Gigi", "Dyson", "unv\_2").  student("Gigi", "Dyson", "unv\_3").    student("Tanya", "Park", Unv):-student("Ann", "Richy", Unv).    teacher("Kristina", "Nikiforova", "physics", "unv\_2").  teacher("Mila", "Kolovanova", "english", "unv\_2").  teacher("Anton", "Popov", "drawing", "unv\_4").    GOAL  %student(Name, Surname, "unv\_1").    %student("Boby", Surname, "unv\_2").    %teacher(\_, \_, "english", "unv\_2").    %teacher(Name, Surname, Subj, "unv\_4").  %teacher("Mila", "Kolovanova", "english", "unv\_2"). |

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | student(Name, Surname, "unv\_1"). | Система сравнивает весь вопрос с первым фактом. В вопросе две переменные, система свяжет их с соответствующими значениями знания (при условии, что природа аргументов одинаковая) и ответит на поставленный вопрос, если ответ да, то в качестве побочного эффекта на экран будут выведены значения переменных. Чтобы продолжить поиск система отменяет это решение (то есть переменная теряет своё значение), и далее продолжается поиск решения. |  |
| 2 | student("Boby", Surname, "unv\_2"). | Аналогично, только в вопросе только одна переменная и две константы. |  |
| 3 | teacher(\_, \_, "english", "unv\_2"). | Аналогично, только разница в том, что не столько важно значение анонимных переменных, сколько важно количество аргументов |  |
| 4 | teacher(Name, Surname, Subj, "unv\_4"). | Аналогично вопросу №1/2 |  |
| 5 | teacher("Mila", "Kolovanova", "english", "unv\_2"). | В вопросе нет переменных, поэтому проверяется только совпадение констант. |  |

**Л/р 13**

**Задание**

Составить программу

**Вопросы**

1. **Что представляет из себя программа на Prolog?**

Программа на Prolog состоит из базы знаний, которую составляют факты и правила.

1. **Структура программы на Prolog?**

Программа на Prolog состоит из разделов, которые начинаются с заголовка.

Разделы:

* Директивы компилятора – зарезервированные символьные константы
* CONSTANTS – раздел описания констант
* DOMAINS – раздел описания доменов
* DATABASE – раздел описания предикатов внутренней базы данных
* PREDICATES – раздел описания предикатов
* CLAUSES – раздел описания предложений базы знаний
* GOAL - раздел описания внутренней цели (вопроса)

В программе могут быть не все разделы.

1. **Как реализуется программа на Prolog?**

Описывается база знаний (факты и правила), задаётся вопрос.

1. **Как формируются результаты работы программы?**

Система пытается найти среди базы знаний такие значения переменных, чтобы ответить «Да» на поставленный вопрос.

1. Назначение и использование переменных.
2. Порядок формирования результата работы программы