|  |  |  |
| --- | --- | --- |
| МИНОБРНАУКИ РОССИИ  Федеральное государственное  бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |
| Институт Информационных Технологий  Кафедра МОСИТ | | |
| **ЛАБОРАТОРНАЯ РАБОТА** | | |
| по дисциплине «**Функциональное и логическое программирование**» | | |
| **Тема лабораторной работы ОСНОВЫ ЯЗЫКА ПРОЛОГ** | | |
| **Студент группы** | ИКБО-11-17 Попова Е.Ю. | *(подпись студента)* |
| **Руководитель работы** | Смольянинова В.А. | *(подпись руководителя)* |
| Москва, 2019 | | |

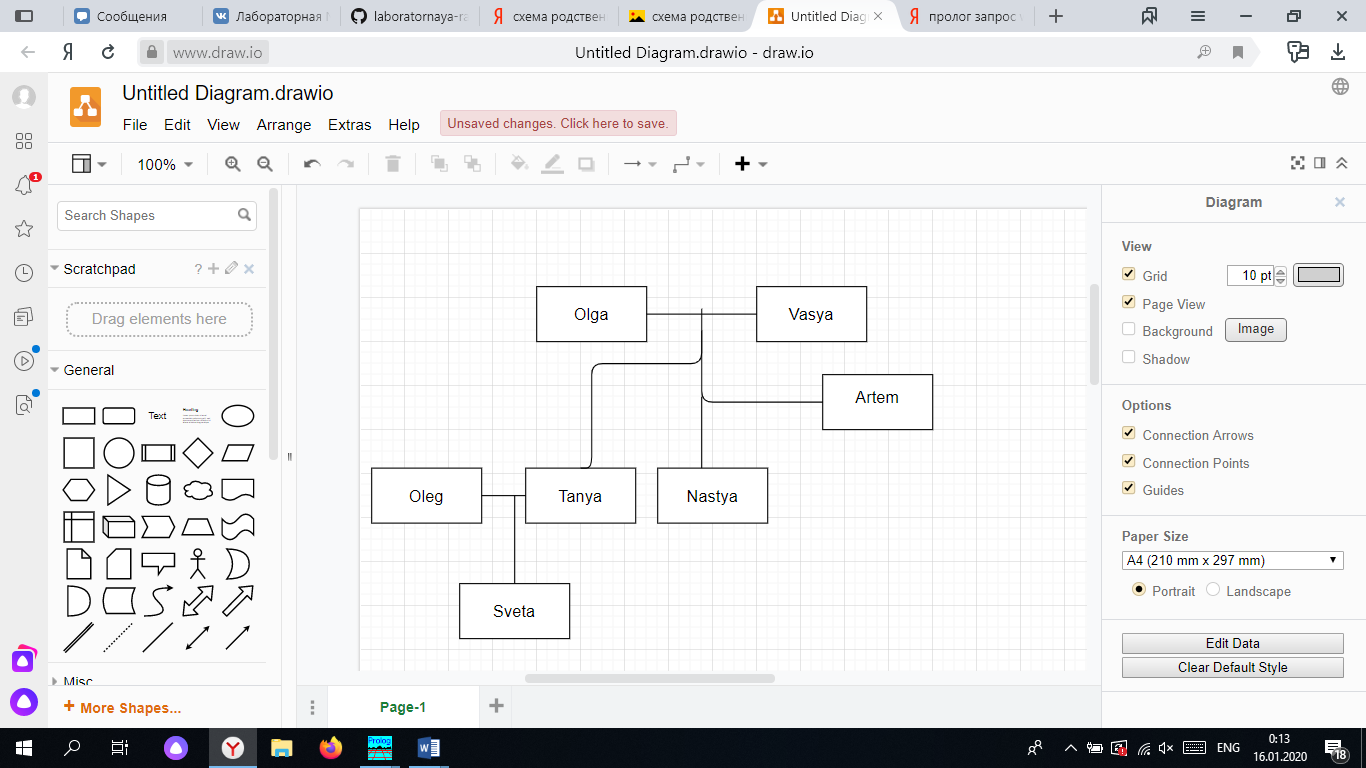
# Постановка задачи

1. Составьте программу Родственные отношения, которая кроме родственных отношений parent (родитель) и ancestor (предок) программа должна содержать одно или несколько из следующих отношений: brother (брат); sister (сестра); grand-father (дедушка); grand-mother (бабушка); aunt (Тетя).
2. Составьте программу, используя отношения likes «нравится») и can\_buy («может купить»).
3. Составьте собственную программу, состоящую из фактов и правил. Проверьте ее работу.

# Ход работы

## 1. Программа Родственные отношения

Схема родственных отношений



Исходный код программы «Родственные отношения»

predicates

male(String)

female(String)

nondeterm parent(String, String)

nondeterm ancestor(String,String)

nondeterm brother(String, String)

nondeterm sister(String, String)

nondeterm grandfather(String, String)

nondeterm grandmother(String, String)

nondeterm aunt(String, String)

clauses

male("Vasya").

male("Artem").

male("Oleg").

female("Olga").

female("Tanya").

female("Sveta").

female("Nastya").

parent("Olga", "Tanya").

parent("Vasya", "Tanya").

parent("Olga", "Nastya").

parent("Vasya", "Nastya").

parent("Olga", "Artem").

parent("Vasya", "Artem").

parent("Oleg","Sveta").

parent("Tanya", "Sveta").

ancestor(X,Z):-parent(X,Z),X<>Z.

ancestor(X,Z):-parent(X,Y),ancestor(Y,Z),X<>Z.

brother(X,Y):-parent(Z,X),parent(Z,Y),male(X),X<>Y.

sister(X,Y):-parent(Z,X),parent(Z,Y),female(X),X<>Y.

grandfather(X,Y):-parent(X,Z),parent(Z,Y),male(X),X<>Y.

grandmother(X,Y):-parent(X,Z),parent(Z,Y),female(X),X<>Y.

aunt(X,Y):-parent(Z,Y),sister(X,Z),female(X),X<>Y. aunt(X,Y):-parent(Z,Y),sister(X,Z),female(X),X<>Y.

Пример запросов:

1. Запрос: sister("Tanya","Artem").

Результат: yes

1. Запрос: grandmother("Olga","Sveta").

Результат: yes

1. Запрос: grandmother("Tanya","Sveta").

Результат: no

1. Запрос: aunt(Who,"Sveta").

Результат: Nastya

## 2. Программа «Нравится/Может купить»

Исходный код

domains

sym=symbol

n=integer

predicates

nondeterm likes(sym,sym)

nondeterm cost(sym,n)

nondeterm have(sym,n)

nondeterm can\_buy(sym,sym)

clauses

likes(tom,cat).

likes(tom,apple).

likes(jane,dress).

likes(jane,cat).

cost(cat,150).

cost(apple,30).

cost(dress,200).

have(tom,100).

have(tom,150).

can\_buy(X,Y):-likes(X,Y),cost(Y,Price),have(X,Money),Price<=Money.

Примеры запросов:

Запрос: can\_buy(tom,cat).

Результат: no

Запрос: can\_buy(tom,cat).

Результат: no

Запрос: can\_buy(jane,cat).

Результат: yes

Запрос: can\_buy(jane,dress).

Результат: no

## Собственная программа – сдача экзамена. Экзамен сдан, если студент пришел на экзамен и вытянул билет, который он знает.

Исходный код программы

domains

sym=symbol

n=integer

predicates

nondeterm went(sym)

nondeterm know\_ticket(sym,n)

nondeterm pull\_ticket(sym,n)

nondeterm pass\_exam(sym)

clauses

went(tom).

went(jane).

know\_ticket(tom,1).

know\_ticket(tom,2).

know\_ticket(tom,3).

know\_ticket(jane,4).

know\_ticket(jane,5).

know\_ticket(jane,6).

pull\_ticket(tom,2).

pull\_ticket(tom,10).

pass\_exam(X):-went(X),know\_ticket(X,Number),pull\_ticket(X,Ticket),Number=Ticket.

Примеры запросов:

Запрос: pass\_exam(tom).

Результат: yes

Запрос: pass\_exam(jane).

Результат: no

Запрос: pass\_exam(olga).

Результат: no

# Выводы

В результате выполнения лабораторной работы были освоены основы языка Prolog.