Оглавление

[Введение 5](#_Toc422236189)

[1 Проектная часть 6](#_Toc422236190)

[1.1 Постановка задачи 6](#_Toc422236191)

[1.1.1 Концептуальный анализ предметной области 7](#_Toc422236192)

[1.1.2 Концептуальная модель предметной области 9](#_Toc422236193)

[1.1.3 Представление задачи в исчислении высказываний 11](#_Toc422236194)

[1.1.4 Представление задачи в исчислении предикатов 13](#_Toc422236197)

[1.1.5 Реализация задач на языке Visual Prolog 5.0 15](#_Toc422236198)

[Вывод 19](#_Toc422236199)

# **Введение**

Основной задачей данной курсовой работы является изучение возможностей логического программирования.

Настоящая работа посвящена логическому программированию. Предметной областью, на основе которой была проведена эта работа, является приготовление торта.

В результате выполнения работы для данной предметной области было необходимо провести: концептуальный анализ, разработать концептуальную модель, представить задачу в исчислении высказываний и предикатов, и реализовать задачи на логическом языке Prolog в среде Visual Prolog 5.

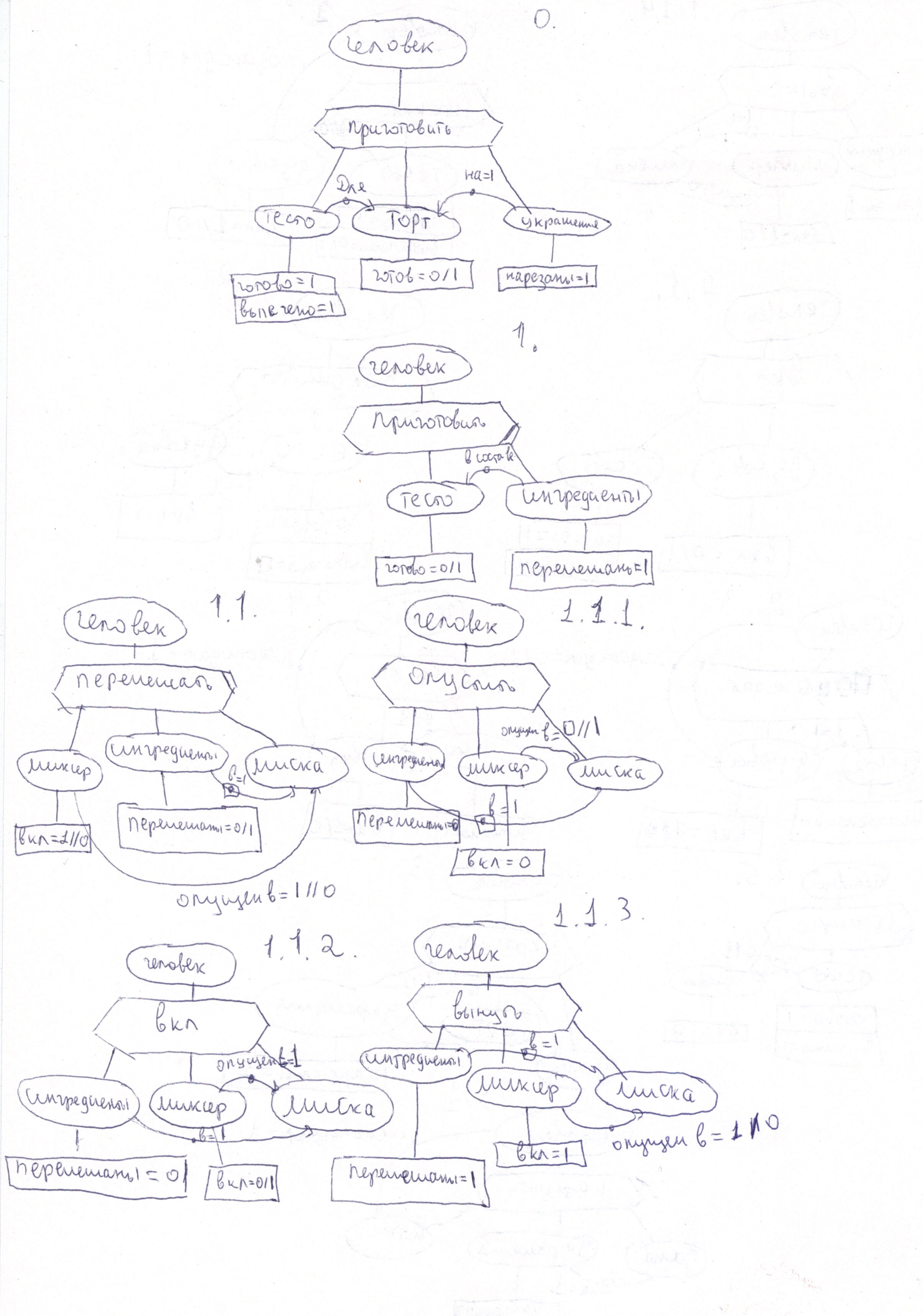
Основной задачей данной курсовой работы является изучение принципов реализации задачи с помощью логических моделей предметного знания.

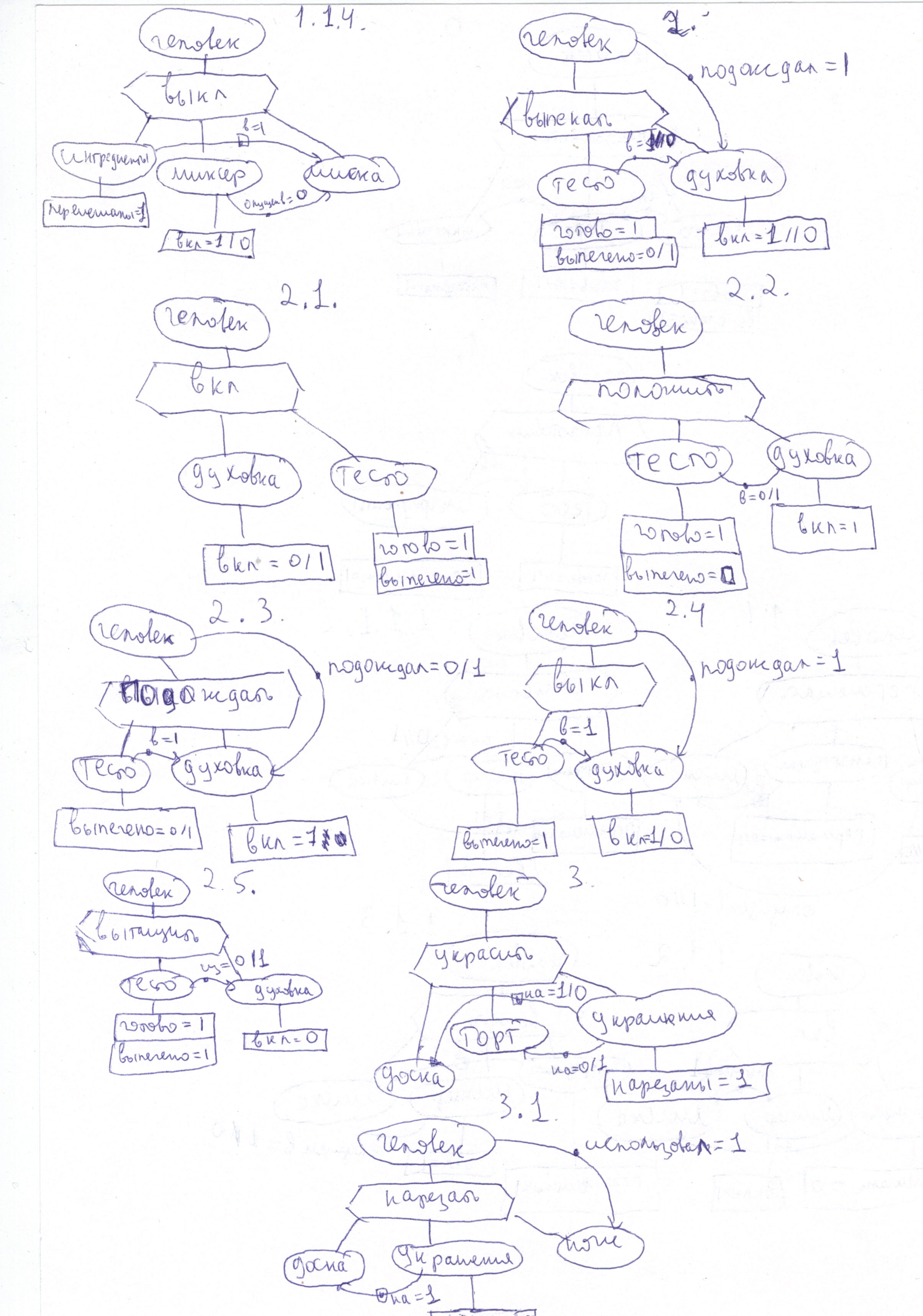
# **Проектная часть**

## Постановка задачи

Торт готов , когда человек положил ингредиенты в миску, перемешал ингредиенты в миске, включил духовку, положил тесто в духовку, подождал, вытащил тесто из духовки, украсил тесто.

### Концептуальный анализ предметной области





### Концептуальная модель предметной области

X - множество объектов предметной области;

C - множество существенных свойств всех объектов;

R - множество существенных отношений между объектами

G - множество допустимых (необходимых) действий над объектами.

G (ЧЕЛОВЕК)= {ПРИГОТОВИТЬ(ТОРТ),

ПРИГОТОВИТ(ТЕСТО),

ПЕРЕМЕШАТЬ(ИНГРЕДИЕНТЫ),

ОПУСТИТЬ(МИКСЕР),

ВКЛЮЧИТЬ(МИКСЕР),

ВЫНУТЬ(МИКСЕР),

ВЫКЛЮЧИТЬ(МИКСЕР),

ВЫПЕКАТЬ(ТЕСТО),

ВКЛЮЧИТЬ(ДУХОВКУ),

ПОЛОЖИТЬ(ТЕСТО),

ПОЖДАТЬ(ДУХОВКУ),

ВЫКЛЮЧИТЬ(ДУХОВКУ),

ВЫТАЩИТЬ(ТЕСТО),

УКРАСИТЬ(ТОРТ),

НАРЕЗАТЬ(УКРАШЕНИЯ)}

X = {ЧЕЛОВЕК,

ТОРТ,

ТЕСТО,

ИНГРЕДИЕНТЫ,

МИКСЕР,

МИСКА,

ДУХОВКА,

ДОСКА,

НОЖ,

УКРАШЕНИ}

С(ТОРТ){ГОТОВ}

С(ТЕСТО){ВЫПЕЧЕНО}

С(УКРАШЕНИЯ){НАРЕЗАНЫ}

С(ДУХОВКА){ВКЛЮЧЕНА}

R = {ДЛЯ(ТЕСТО,ТОРТ),

НА(УКРАШЕНИЯ,ТОРТ),

В\_СОСТАВЕ(ИНГРЕДИЕНТЫ,ТЕСТА),

В(МИСКА,ИНГРЕДИЕНТЫ),

ОПУЩЕН\_В(МИКСЕР,МИСКА),

ПОЛОЖЕНО\_В(ТЕСТО,ДУХОВКА),

ПОДОЖДАТЬ(ЧЕЛОВЕК,ДУХОВКА),

ВЫТАЩИТЬ\_ИЗ(ДУХОВКА,ТЕСТО),

НА(УКРАШЕНИЯ,ДОСКА),

ИСПОЛЬЗОВАЛ(ЧЕЛОВЕК,НОЖ)}

### Представление задачи в исчислении высказываний

*Алфавит*

1. Торт готов
2. Человек положил ингредиенты в миску
3. Человек перемешал ингредиенты в миске (уже)
4. Человек перемешивает ингредиенты в миске (сейчас)
5. Включил духовку
6. Человек положил тесто в духовку
7. Подождал
8. Вытащил тесто из духовки
9. Человек украшает тесто

*Аксиомы*

1. a → f i = (a ˅ f) (a ˅ i)
2. f → c b = (f ˅ c) (f ˅ b)
3. d → b c = (d ˅ b) (d ˅ c)
4. c → e = (c ˅ e)
5. h → f c g = (h ˅ f) (h ˅ c) (h ˅ g)

*I этап: докажем на противоречивость аксиомы*

(1,3) = (a ˅ c)

(1,4) = (a ˅ b)

(3,6) = (f ˅ d)

(3,7) = (e ˅ f)

(6,9) = (d ˅ h)

(8,3) = (h ˅ c)

(8,4) = (h ˅ b)

■ не существует → система не противоречива

I I этап: доказываем выражение e → a методом опровержения с помощью метода резолюции

e → a

e → a = e ˅ a = e a

(19,7) = c

(20, 11) = c

(21,22) = ■ – существует → e → a - верно

* + 1. Представление задачи в исчислении предикатов

1. Приготовил (X,Y)
2. В(М,И)
3. Перемешано\_в (И,M)
4. Перемешивает\_в (И,М)
5. Вкл (Х, Д)
6. Положил\_в (Т, Д)
7. Подождал (Х, Д)
8. Вытащил\_из (Х,Т,Д)
9. Украсил (Х,У)

ⱯхⱯу (Приготовил (X,Y) → Украсил (Х,У) & Вытащил\_из (Х,Т,Д))

ⱯхⱯу (Украсил (Х,У) → Перемешано\_в (И,M) & В(М,И))

Перемешивает\_в (И,М) → В(М,И) & Перемешано\_в (И,M)

Ɐх (Перемешано\_в (И,M) → Вкл (Х, Д) )

Ɐх (Вытащил\_из(Х,Т,Д)→Подождал(Х,Д)&Вкл(Х,Д) &Положил\_в(Т,Д))

1. (Приготовил (X,Y) ˅ Украсил (Х,У)) (Приготовил (X,Y) ˅ Вытащил\_из (Х,Т,Д))
2. (Украсил (Х,У) ˅ Перемешано\_в (И,M)) (Украсил (Х,У) ˅ В(М,И))
3. (Перемешивает\_в (И,М) ˅В(М,И))(Перемешивает\_в (И,М) ˅ Перемешано\_в (И,M))
4. (Перемешано\_в (И,M) ˅ Вкл (Х, Д))
5. (Вытащил\_из(Х,Т,Д)˅Положил\_в(Т,Д))(Вытащил\_из(Х,Т,Д)˅Вкл(Х,Д)) (Вытащил\_из(Х,Т,Д) ˅ Подождал(Х,Д))

Докажем выражение ⱯхⱯуВкл(Х,Д)→Приготовил(X,Y) методом опровержения с помощью метода резолюции

ⱯхⱯуВкл(Х,Д)→Приготовил(X,Y)

Доказательство:

Шаг 1 берем отрицание

ⱯхⱯуВкл(Х,Д)→Приготовил(X,Y)

Шаг 2 преобразование

х\existsу (Вкл(Х,Д)→Приготовил(X,Y)) = \existsх\existsу(Вкл(Х,Д)&Приготовил(X,Y))



Шаг 3 заменяем Х на Повар(П) и заменяем У на Торт(То), убираем \existsх\existsу

Вкл(П,Д)& Приготовил(П,То)

(1,3): Приготовил(X,Y) ˅ Перемешано\_в (И,M)

В 13 Заменяем Х на Повар(П) и У на Торт(То)

(13,12): Перемешано\_в(И,М)

(11,7): Перемемано\_в(И,М)

(14,15) : ■ - существует → ⱯхⱯуВкл(Х,Д)→Приготовил(X,Y) - верно

### Реализация задач на языке Visual Prolog 5.0

#### Перевод исчисление предикатов на язык Visual Prolog

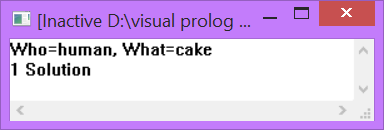
**Код программы**

domains  
Things = plate;oven;dough;ingredients  
X = human  
Y = cake  
  
predicates  
  
 cooked(X,Y) % приготовил  
 in(Things,Things) % находится\_в  
 mixed(Things,Things) % перемешано  
 turn\_on(X,Things) % включено  
 put\_in(X,Things,Things) % положено  
 wait(X) % подождал  
 baked(Things,Y) % выпечено  
 pulled\_out(X,Things,Things) %вытащено  
 decorated(X,Things) %украшено  
  
clauses  
 cooked(human,cake):- baked(dough,cake),in(plate,ingredients),put\_in(human,ingredients,oven).  
 decorated(human,dough):- pulled\_out(X,dough,oven), baked(dough,Y), wait(X).  
 put\_in(human,ingredients,oven):-  mixed(ingredients,plate),  in(plate,ingredients).  
 turn\_on(human,oven):- put\_in(X,ingredients,oven),mixed(ingredients,plate),in(ingredients,plate).  
 baked(dough,cake):- wait(X),turn\_on(X,oven),in(plate,ingredients),mixed(ingredien,plate).  
 wait(human):- turn\_on(X,oven),mixed(ingredients,plate).  
 mixed(ingredients,plate):-in(plate,ingredients).  
 pulled\_out(human,dough,oven):- mixed(ingredients,plate),baked(dough,Y).  
 in(plate,ingredients).  
  
  
  
goal  
%mixed(What,Who). % перемешал что кто  
%in(Where, What). % Где Что находится   
%baked(dough,Y). % Выпечено тесто для Торта   
cooked(Who,What).% Приготовлено Кем ? Что?

Результат

goal

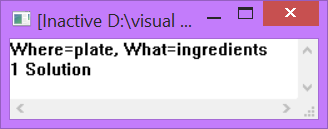
cooked(Who,What). % Приготовлено Кем ? Что?



Результат

goal

in(Where, What). % Где Что находится



#### 1.1.5.2 Реализация на Visual Prolog 5.0 действий, описанных в исчислении предикатов, с помощью (функционального терма state) и предикатов Act и CanAct.

**Код программы**

domains   
 Fact = mixed;not\_mixed;baked;not\_baked;decorated;not\_decorated;

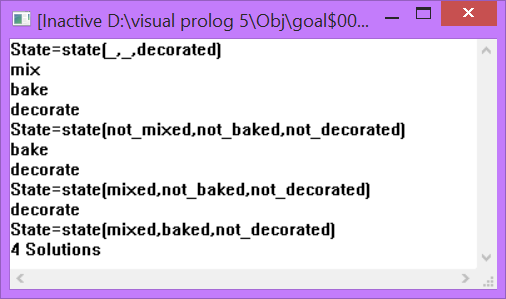
% (перемешано/не перемешано; выпечено/не выпечено; украшено/не украшено)  
 state (Fact,Fact,Fact) %состояния  
 Action = mix;bake;decorate % факты   
   
predicates  
act (Fact, Action, Fact)  
canAct (Fact)  
   
clauses  
act(state(not\_mixed, not\_baked, not\_decorated), mix,  
 state(mixed,not\_baked,not\_decorated)):- write(mix),nl.  
act(state(mixed,not\_baked,not\_decorated), bake,  
 state(mixed,baked,not\_decorated)):- write(bake),nl.  
act(state(mixed,baked,not\_decorated),decorate,  
 state(mixed,baked,decorated)):-write(decorate),nl.  
canAct(state(\_,\_,decorated)).  
canAct(State1):- act(State1, Act, State2), canAct(State2).

goal  
  %canAct(state(not\_mixed,not\_baked,decorated)).  
%canAct(state(mixed,baked,not\_decorated)).  
%canAct(state(mixed,not\_baked,decorated)).  
%canAct(state(mixed,not\_baked,not\_decorated)).  
canAct (State).

Результат

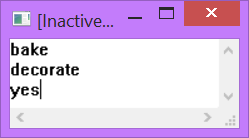
goal

canAct (State). % описывает все состояния



goal

canAct(state(mixed,not\_baked,not\_decorated)). % описывает что (перемешано, невыпечено, неукрашено )



# **Вывод**

В ходе выполнения курсовой работы был получен практический навык в разработке концептуальной модели предметной области и реализации ее на языке Visual Prolog 5.0.