Министерство науки и высшего образования РФ

Федеральное государственное бюджетное образовательное

учреждение высшего образования

«Уфимский университет науки и технологий»

Факультет информатики и робототехники

Кафедра вычислительной математики и кибернетики

Отчет к практической работе №1

по дисциплине «Функциональное и логическое программирование»

Выполнил:

студент группы МО-321 Шемануев А.Е.

Проверил:

Ризванов Д. А.

Уфа, 2022

**Задание:**

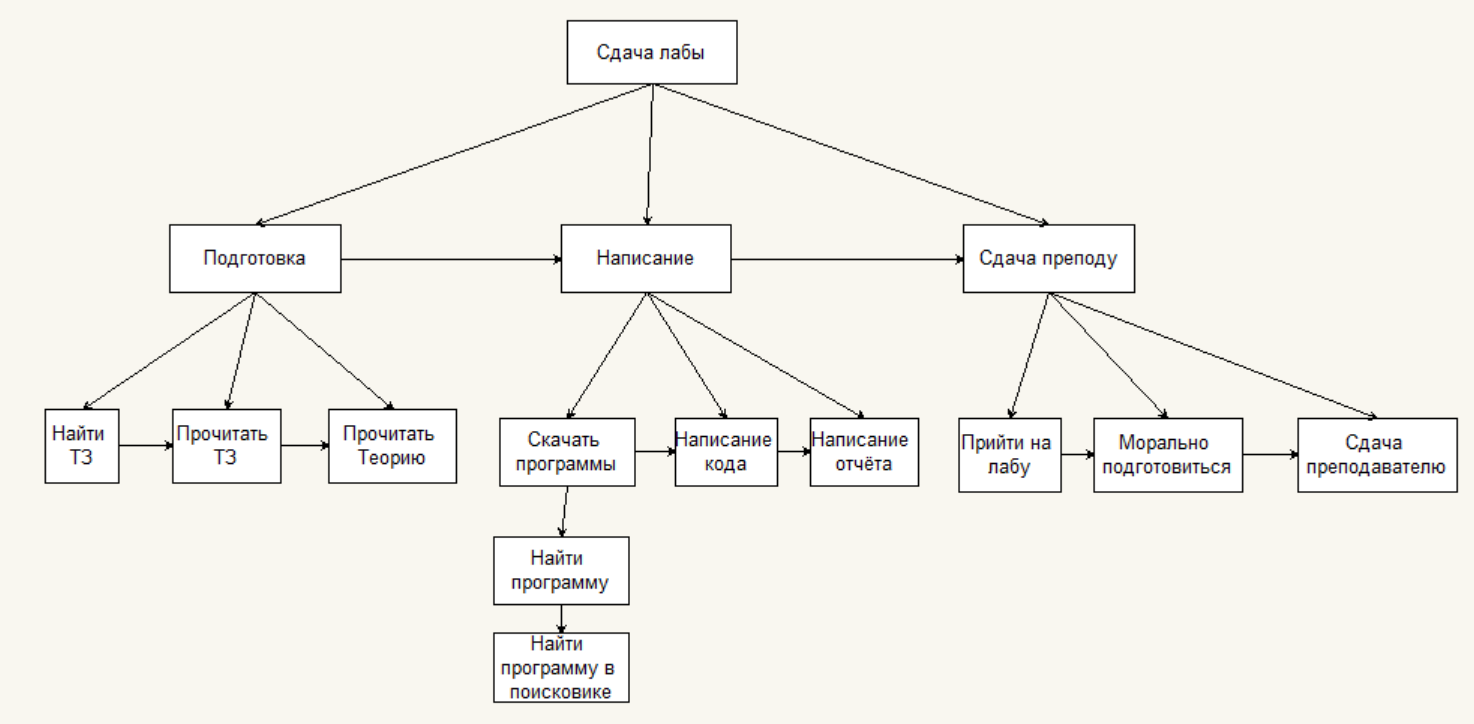
Построить модель предметной области по выбору, описывающую какой-либо процесс. Провести декомпозицию модели.

Написать на Прологе программу, описывающую предметную область и характер взаимосвязи процессов между собой. Реализовать в программе отношения иерархии или включения (например, процесс *A* включает в себя работы *A1*, *A2*, *A3*) и последовательности или предшествования (например, процесс *A2* следует за процессом *A2*).

Программа должна предоставлять возможность получения ответа на следующие вопросы:

1. какие атомарные работы осталось выполнить до завершения всего процесса, если указана текущая завершенная работа;
2. какие атомарные работы были уже выполнены, если указана текущая завершенная работа;
3. какие атомарные работы расположены между двумя другими работами (необязательно атомарными).

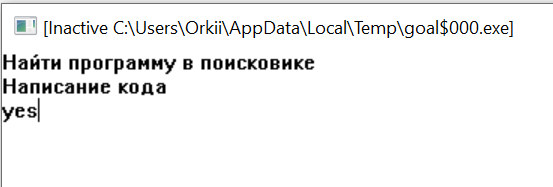
**Когнитивная модель:**

****

**Код программы:**

|  |
| --- |
| domains  i = integer  s = string  predicates  nondeterm nextT(s, s) %2 is next for 1 %Для родителей  nondeterm nextU(s, s) %Go up  nondeterm nextD(s, s) %Go down  nondeterm next(s, s)    nondeterm prev(s, s)  nondeterm prevU(s, s)  nondeterm prevD(s, s)    nondeterm parent(s, s)    nondeterm left(s)  nondeterm done(s)  nondeterm doneGD(s)  nondeterm doneGR(s)    nondeterm between(s,s)    nondeterm firstOfBranch(s,s)  nondeterm lastOfBranch(s,s)    nondeterm lastOfAll(s,s)  nondeterm lastOfAll2(s,s)  nondeterm lastOfAll3(s,s)    nondeterm firstOfAll(s,s)  nondeterm firstOfAll2(s,s)  nondeterm firstOfAll3(s,s)  nondeterm atom(s)      clauses  nextT("Подготовка", "Написание").  nextT("Написание", "Сдача преподу").    nextT("Найти ТЗ", "Прочитать ТЗ").  nextT("Прочитать ТЗ", "Прочитать теорию").    nextT("Скачать программы", "Написание кода").  nextT("Написание кода", "Написание отчёта").    nextT("Прийти на лабу", "Морально подготовиться").  nextT("Морально подготовиться", "Сдача преподавателю").    next(A, T):- nextT(A, B), atom(B), B=T, !.  next(A, T):- nextT(A, B), nextD(B,T), !.  next(A, T):- nextU(A, T), !.    nextU(A,T):- parent(B,A), nextT(B,C), nextD(C,T).  nextU(A,T):- parent(B,A), nextU(B,T).    nextD(A,T):- atom(A), A=T, !.  nextD(A,T):- parent(A,B), firstOfBranch(B,C), nextD(C,T).      prev(A, T):- nextT(B, A), atom(B), B=T, !.  prev(A, T):- nextT(B, A), prevD(B,T), !.  prev(A, T):- prevU(A, T), !.    prevU(A,T):- parent(B,A), nextT(C,B), prevD(C,T).  prevU(A,T):- parent(B,A), prevU(B,T).    prevD(A,T):- atom(A), A=T, !.  prevD(A,T):- parent(A,B), lastOfBranch(B,C), prevD(C,T).    parent("Сдача лабы", "Подготовка").  parent("Сдача лабы", "Написание").  parent("Сдача лабы", "Сдача преподу").    parent("Подготовка", "Найти ТЗ").  parent("Подготовка", "Прочитать ТЗ").  parent("Подготовка", "Прочитать теорию").    parent("Написание", "Скачать программы").  parent("Написание", "Написание кода").  parent("Написание", "Написание отчёта").    parent("Сдача преподу", "Прийти на лабу").  parent("Сдача преподу", "Морально подготовиться").  parent("Сдача преподу", "Сдача преподавателю").    parent("Скачать программы", "Найти программу").  parent("Найти программу", "Найти программу в поисковике").  firstOfBranch(A,B):- nextT(C,A), firstOfBranch(C,B).  firstOfBranch(A,B):- B = A.    lastOfBranch(A,B):- nextT(A,C), lastOfBranch(C,B).  lastOfBranch(A,B):- B = A.    left(A):- lastOfAll(A, T), A=T, !.  left(A):- next(A, B), write(B), nl, left(B).    %done(A):- parent(A,B), done(B).  done(A):- firstOfAll(A,C), write(C), nl, between(C,A).    %done(A):- doneGR(A).    doneGD(A):- parent(A,B), firstOfBranch(B,C), doneGD(C).  doneGD(A):- write(A), nl, doneGR(A).    doneGR(A):- lastOfAll(A, T), A=T, !.  doneGR(A):- next(A, B), write(B), nl, doneGR(B).    between(A,T):- prev(T,C),A=C, !.  between(A,T):- next(A,B), write(B), nl, between(B,T).  %between(A,T):- parent(A,B), next(A, C), parent(C, D), between(D,T).  %between(A,T):- write(A), nl, next(A, B), between(B,T).  %between(A,T):- parent(B,A), between(B,T).    lastOfAll(A,T):- parent(A, B), lastOfAll(B,T), !.  lastOfAll(A,T):- lastOfAll2(A,T), !.    lastOfAll2(A,T):- parent(B,A), next(B, C), parent(C,E), lastOfAll2(E,T).  lastOfAll2(A,T):- lastOfAll3(A,T), !.    lastOfAll3(A,T):- next(A, B), lastOfAll3(B,T).  lastOfAll3(A,T):- A=T, !.        firstOfAll(A,T):- parent(A, B), firstOfAll(B,T), !.  firstOfAll(A,T):- firstOfAll2(A,T), !.    firstOfAll2(A,T):- parent(B,A), prev(B, C), parent(C,E), firstOfAll2(E,T).  firstOfAll2(A,T):- firstOfAll3(A,T), !.    firstOfAll3(A,T):- prev(A, B), firstOfAll3(B,T).  firstOfAll3(A,T):- A=T, !.    atom(A):- parent(A,B), !, fail.  atom(\_).  goal  %parent("Сдача преподу", A).    %lastOfAll("Написание",T),  %firstOfAll("Написание",A).      %done("Написание кода").  %done("Найти программу").  %done("Написание").    %left("Сдача лабы").  between("Подготовка", "Написание отчёта"). |

**Примеры выполнения:**

****

**Вывод:** В ходе практической работы была Построена модель предметной области по выбору, описывающую какой-либо процесс. Провести декомпозицию модели.

:

1. какие атомарные работы осталось выполнить до завершения всего процесса, если указана текущая завершенная работа;
2. какие атомарные работы были уже выполнены, если указана текущая завершенная работа;
3. какие атомарные работы расположены между двумя другими работами (необязательно атомарными).