# LP\_Capstone

## Курсовой проект по курсу «Логическое программирование»

### Цель работы

\* Применить навыки логического и функционального программирования при решении реалистичной задачи

\* Построить и проанализировать родословное дерево своего рода на несколько поколений

### Задание

1. Создать родословное дерево своего рода на несколько поколений (3-4) назад в стандартном формате GEDCOM. Для этого рекомендуется использовать сервис MyHeritage.com, зарегистрировавшись в нем и подключив к заполнению дерева других родственников (указав их контактный e-mail при регистрации, что приведет к автоматической рассылке приглашения на их адрес). Если Вы по каким-то причинам не хотите использовать свою родословную - вы можете использовать файл [родословной европейской знати](http://www.rusgenealog.ru/gedcom/royal\_gen.zip).

2. Преобразовать файл в формате GEDCOM в набор утверждений на языке Prolog, используя одно из следующих представлений в зависимости от номера в группе в соответствии с формулой ((N-1)+3) mod 4 + 1:

1. с использованием предиката `parents(потомок, отец, мать)`

2. с использованием предикатов `parent(родитель, ребенок)`, `sex(человек, m/f)`

3. с использованием предиката `child(ребенок, родитель)`, `male(человек)`, `female(человек)`

4. с использованием предикатов `father(отец, потомок)` и `mother(мать, потомок)`

Для преобразование рекомендуется использовать один из следующих инструментов (в порядке уменьшение предпочтительности):

\* Функциональный язык программирования (F#, Haskell, LISP, …)

\* Язык Prolog

\* Языки высокого уровня, позволяющие легко проводить обработку текста (Python, Ruby, Perl, awk, …)

\* Другие императивные и ОО языки программирования

> Отличная оценка подразумевает использование одного из первых двух пунктов. Использование неадекватного инструмента, приводящего к чрезмерно сложной программе, не рекомендуется, и оценивается соответственно.

3. Реализовать предикат проверки/поиска следующих родственников в соответствии с номером варианта ((N-1) mod 7)+1:

1. Шурин

2. Деверь

3. Золовка

4. Двоюродный брат

5. Свекровь

6. Теща

7. Троюродный брат или сестра

4. [На оценки хорошо и отлично] Реализовать программу на языке Prolog, которая позволит определять степень родства двух произвольных индивидуумов в дереве, например:

```prolog

?- relative(brother, ‘Петя’, X).

X = ‘Вася’

?- relative(W,’Петя’,’Вася’).

X = brother

X = child – father

% т.е. Вася является ребенком отца Пети

X = child – mother

% т.е. Вася является ребенком матери Пети

```

5. [На оценки хорошо и отлично] Реализовать естественно-языковый интерфейс к системе, позволяющий задавать вопросы относительно степеней родства, и получать осмысленные ответы. Допускается использовать английский язык в качестве базового. Для отличной оценки требуется:

\* Учесть контекст (возможность использования it/him/her с отсылкой на упоянутый на прошлом шаге объект)

\* Выполнять запросы относительно количества (How many brothers does Mary have?)

\* Выполнить разбор предложения с построением смысловой модели.

### Оформление отчета

Отчет по курсовому проекту должен содержать:

1. ОРИГИНАЛЬНЫЙ мини-реферат `Essay.md` (объемом более 3000 слов) на одну из следующих тем, в соответствии с номером варианта (N mod 13)+1:

1. Как научить младшую сестру/брата логическому программированию

2. Логическое программирование при создании современных информационных систем

3. Как научить вашу бабушку/дедушку логическому программированию

4. Как использовать логические языки чтобы научить компьютер играть в шахматы/шашки

5. Логические языки как первые языки для обучения программированию

6. Логические языки как путь к автоматическому решению задач компьютером

7. Логические языки и базы данных

8. Типовые и бестиповые логические языки: обзор и сравнение.

9. Современные языки и системы логического программирования

10. Математическая логика и логическое программирование: сравнение

11. Что было бы, если бы человечество придумало Prolog как первый язык программирования?

12. Почему не получилось реализовать ЭВМ пятого поколения?

13. Сравнение реализации алгоритмов поиска на логических и императивных языках

14. [Иванопуло] За что я люблю логическое программирование

Реферат должен быть написан на основе прочтения 3-5 источников (указанных в конце в списке литературы), и представлять собой оригинальный текст, написанный Вами лично. Все дословные заимствования должны быть оформлены в виде цитат. Обязательно наличие в реферате умных мыслей.

Тему реферата можно предложить самостоятельно. Для этого необходимо:

- создать pull request (в репозитории [LP\_Capstone](http://github.com/mailabs-education-lp/LP\_Capstone)) к этому документу с предложенной новой темой реферата (в конце темы напишите в скобках свою фамилию)

, - после того, как pull request одобрен и тема стала доступна в репозитории, Вы можете брать эту тему и писать реферат

2. Отчет по курсовому проекту `Report.md`

3. Все исходные файлы, необходимые для запуска проекта

### Плагиат

Плагиат при выполнении курсового проекта не допускается:

1. Дословное заимствование текста реферата (более 10% текста) из каких-либо источников или друг у друга.

2. Использование целиком фрагментов программ других учащихся или из доступных источников

При обнаружении плагиата работа аннулируется, и учащемуся выдается новый вариант задания, при этом впоследствии выполненная работа не может быть оценена выше, чем удовлетворительно.

Отнеситесь к этому серьезно, это важно!