**Уровень 1.**

**Задание 1.**

Для выполнения этого задания я решила написать бота, который продаёт фрукты.

<category>  
   <pattern>Здравствуйте</pattern>  
   <template>Добрый день!</template>  
</category>

<category>

<pattern>Погода сегодня \*</pattern>

<template><srei>Да</srei>, <star/></template>

</category>

<category>

<pattern>Можете мне что-то посоветовать?</pattern>

<template>

<random>  
         <li>У нас есть вкусные бананы</li>  
         <li>Я могу предложить вам яблоки</li>  
      </random>

</template>

</category>

<category>

<pattern>А у вас есть \*</pattern>

<template>Да, у нас есть<star/></template>

</category>

<category>

<pattern>По чём у вас \*</pattern>

<template><srai>Сегодня цены на все фрукты 1000 рублей за кг</srai>

</template>

</category>

<category>

<pattern> \* отдадите за \*</pattern>

<template><srai>Нет</srai></template>

</category>

<category>

<pattern>Хорошо, вы теряете покупателя</pattern>

<template>Подождите</template>

</category>

<category>

<pattern>Отдадите персики за \*</pattern>

<template><sr/></template>

</category>

<category>

<pattern>Спасибо что пошли навстречу</pattern>

<template>Это только за ваши красивые глаза</template>

</category>

<category>

<pattern>До свидания</pattern>

<template>

<random>  
         <li>До свидания, хорошего дня</li>  
         <li>Приходите ещё</li>  
</random>

</template>

</category>

**Задание 2.**

**1.** Для работы AIML-бота необходимо 2 компонента: файл AIML и программа. Файл AIML - содержит набор категорий (category). Категория открывается тегом <category> и закрывается тегом </category>. Категория включает паттерны (pattern) и темплейты (template). Категория может включать только один паттерн и один темплейт. Паттерн и темплейт должны находиться внутри категории. Паттерн пишется большими буквами. Для определения произвольного слова или группы слов используется знак \*. Темплейт может включать одну реплику или несколько реплик. В этом случае необходимо использовать дополнительный тег <random>.

В AIML можно создавать ссылки из темплейтов на паттерны. Делается это с помощью тега <srai>.

<star/> - это та самая звездочка, которая стоит в паттерне. Таким образом то, что находится после "Поговорим о" будет вставлено в ответную реплику.

<sr/> это сокращенная запись <srai><star/></srai>, то есть ссылка на звездочку.  
Разговорная речь, как правило, содержит некоторое количество т.н. слов-паразитов, которые человеческий мозг воспринимает без проблем, в то время как для мозга бота, написанного на AIML, такие слова могут стать серьезной проблемой. Дополнять каждую конструкцию дополнительным паттерном для отсечения этих слов-паразитов слишком трудоемко. Поэтому гораздо проще сразу "обрезать" эти слова, как будто их просто не было:

В настройках бота у вас есть раздел AIML TOPICS:   
и окошко New topic. Вводим в этом окошке название темы, например, Default (потом темы можете добавлять). Темы - это по сути куски AIML для бота.

Для того, чтобы бот не только разговаривал, но и выражал эмоции в ответ на реплики собеседника стандарт AIML дополнен тегом <action>.

**2.** AIML — язык разметки для искусственного интеллекта. Это один из самых первых, и самых простых языков, которые используется для создания ботов.

Первым виртуальным собеседником была программа Элиза, разработанная в 1966 году. Её создатель, Джозеф Вейценбаум. AIML разрабатывался Ричардом Уолессом — инициатором проекта ALICE. Для работы AIML-бота необходимо 2 компонента: файл AIML и программа (интерпретатор), которая обрабатывает входящие реплики собеседника и формирует ответную реплику в соответствии с файлом AIML.

<Set> тег используется для установки значения в переменной. Тег <Get> используется для получения значения из переменной.

Тег <that> используется в AIML для реагирования по контексту

Тег <topic> используется в AIML для хранения контекста, чтобы позже разговор мог быть продолжен на основе этого контекста. Как правило, тег <topic> используется в Да/Нет типе разговора.

Тег <think> позволяет задавать значения переменным, сохраняя их в памяти.

Тег <condition> похож на переключение высказывания в языке программирования. Это помогает реагировать на соответствующие входные данные.

Тег <learnf> позволяет в автоматическом режиме создавать шаблон.

**Уровень 2.**

<https://t.me/LetsayabookBot>



