Прототип 5. Алгоритм парсера

Текстовое описание алгоритма

В документе описан алгоритм парсера (анализатора вводимой игроком команды) и структура данных, которые он использует.

Contents

[Структура данных 2](#_Toc256381670)

[Типы данных 2](#_Toc256381671)

[Объектная модель (ОМ) 2](#_Toc256381672)

[Действия (actions) 2](#_Toc256381673)

[Словарь (dictionary) 2](#_Toc256381674)

[Входные параметры 3](#_Toc256381675)

[Выходные параметры 3](#_Toc256381676)

[Алгоритм 4](#_Toc256381677)

[Основной алгоритм 4](#_Toc256381678)

[Используемые структуры данных 4](#_Toc256381679)

[Алгоритм 4](#_Toc256381680)

[Использует алгоритмы 5](#_Toc256381681)

[Алгоритм «Сопоставление фразы с шаблоном» 5](#_Toc256381682)

[Входные параметры 5](#_Toc256381683)

[Выходные параметры 5](#_Toc256381684)

[Алгоритм 5](#_Toc256381685)

[Алгоритм «Сопоставление ЭлементаШаблона с началом фразы» 5](#_Toc256381686)

[Входные параметры 5](#_Toc256381687)

[Выходные параметры 6](#_Toc256381688)

[Алгоритм 6](#_Toc256381689)

[Алгоритм «Сопоставление синонима с началом фразы» 6](#_Toc256381690)

[Входные параметры 6](#_Toc256381691)

[Выходные параметры 6](#_Toc256381692)

[Алгоритм 6](#_Toc256381693)

[Алгоритм «Сопоставления слова фразы с объектом» 6](#_Toc256381694)

[Входные параметры 6](#_Toc256381695)

[Выходные параметры 6](#_Toc256381696)

[Алгоритм 6](#_Toc256381697)

[Алгоритм «Очистка списка ПодходящихШаблонов от недоступных объектов» 6](#_Toc256381698)

[Входные параметры 6](#_Toc256381699)

[Выходные параметры 6](#_Toc256381700)

[Алгоритм 6](#_Toc256381701)

# Структура данных

В этом разделе описана структура данных, используемых в алгоритме – как имеющихся в системе (объектная модель), так и тех данных, которые подаются на вход алгоритма и получаются на его выходе.

## Типы данных

* Падеж: ИП, РП, ДП, ВП, ТП, ПП
* ВызовПроцедуры:
  + Процедура: ссылка на Процедуру
  + Параметры: массив Чисел

## Объектная модель (ОМ)

Данные из объектной модели системы, используемые в алгоритме.

### Действия (actions)

Действия хранят шаблоны фраз и ссылку на вызов процедуры реакции.

### Словарь (dictionary)

Словарь хранит записи о словах по падежам.

## Входные параметры

## Выходные параметры

# Алгоритм

## Основной алгоритм

### Используемые структуры данных

### Алгоритм

1. Если действий в системе нет, выкидывает исключение «Системная ошибка: нет ни одного действия».
2. Формирует внутренне представление Фразы – массива строк, бья текстовую фразу по разделителям (пробелы)
3. Перебирает все действия в ОМ, для каждого действия сопоставляет фразу с шаблоном, формируя массив всех вариантов (ПодходящиеШаблоны) – в том числе, если объектов на одно место претендует множество, они все перечисляются списком на этом месте.
   1. Ловит все исключения при сопоставлении фраз (которые выкидывает процедура соспоставления), сохраняет последнее. Если это было IFML2ParsePhraseTooLong, формирует по нему обычное исключение, добавляя массив в конце, прокидывает наверх.
4. Если в итоге список получился пустым, выкидывает сохранённое исключение (как причину, что фраза не подошла).
5. Далее проверяет доступность всех объектов, упомнятых во фразе – после проверки получается новый массив ПодходящихШаблонов – уже только с доступными объектами.
   1. Ловит все исключения при проверке доступности объектов (которые выкидывает процедура проверки), сохраняет последнее.
6. Если в итоге список получился пустым, выкидывает сохранённое исключение (как причину, что фраза подошла, но какой-то объект недоступен).
7. Берёт первую запись из получившегося списка. Если в ней есть объекты с несколькими вариантами, выкидывает исключение «Не понятно, который <ИП> имеется в виду». (*Вопрос от парсера с уточнением, какой объект упоминается, будет в следующих прототипах.*)
8. Возвращает ссылку на ВызовПроцедуры и массив формальных параметров, которые сформировались при сопоставлении фразы и шаблона.

### Использует алгоритмы

* Сопоставление фразы с шаблоном
* Очистка списка ПодходящихШаблонов от недоступных объектов

## Алгоритм «Сопоставление фразы с шаблоном»

### Входные параметры

* Фраза – Массив строк
* Шаблон

### Выходные параметры

* ПодходящиеФормальныеЭлементы

### Алгоритм

1. Копирует входящие Фразу и Шаблон в свои переменные для работы
2. Берёт первый элемент Шаблона, пытается его сопоставить с началом фразы. При неудаче исключение пробрасывается наверх, при удаче выполнение алгоритма продолжается
3. Подготавливает выходной параметр ПодходящиеФормальныеЭлементы – создаёт и добавляет в него ПодходящийФормальныйЭлемент, которые вернула процедура сопоставления
4. Отрезает от переменной Шаблона первый элемент, а от переменной Фразы то кол-во элементов, что вернула процедура сопоставления
5. Если в переменных Шаблона и Фразы ничего не осталось – возвращает подготовленный выходной параметр ПодходящиеФормальныеЭлементы
6. Если в переменной Шаблона что-то ещё есть, а в перемнной Фразы – ничего – выбрасывает исключение «<переменная Шаблона, где вместо объектов идут вопросительные слова “что” в падеже элемента, а из синонимов литералов выбирается случайный>?» (*Вопрос от парсера с уточнением «к чему вы хотите привязать лошидь?» будут в следующих прототипах*)
7. Если в переменной Шаблона ничего не осталось, а в переменной Фразы что-то есть – выбрасывает исключение специального типа IFML2ParsePhraseTooLong с текстом «Я бы понял фразу, если бы вы сказали: » с массивом, содержащим ПодходящийФормальныйЭлемент
8. (Иначе) Рекурсивно вызывает сам себя с параметрами: переменная Шаблона, переменная Фразы. ПодходящиеФормальныеЭлементы, которые вернулись после вызова, добавляет к своему выходному параметру.  
   Если при вызове выбросилось исключение:
   1. IFML2ParsePhraseTooLong – вставляет в начало его массива ПодходящийФормальныйЭлемент, пробрасывает наверх
   2. Другие исключения пробрасывает наверх

## Алгоритм «Сопоставление ЭлементаШаблона с началом фразы»

### Входные параметры

* ЭлементШаблона
* Фраза – Массив строк

### Выходные параметры

* ПодходящийФормальныйЭлемент
* Кол-во слов, использованных из Фразы

### Алгоритм

1. Если ЭлементШаблона – литерал, то:
   1. Перебирает его синонимы, пытаясь сопоставить синоним с началом фразы. Ловит все исключения, сохраняет последнее:
      1. Первый удачный (без исключения) кладёт в исходящий параметр ПодходящийФормальныйЭлемент, также заполняет исходящий параметр – Кол-во слов. Возвращает управление выше.
   2. Если не было ни одного удачного сопоставления, прокидывает наверх сохранённое исключение. Если не было ни одного удачного, ни одного исключения, выкидывает исключение: «Системная ошибка: при сопоставлении ЭлементаШаблона с началом фразы проверка синонимов ни к чему не привела».
2. Если ЭлементШаблона – объект, то:
   1. Пытается найти объект по первому слову фразы:
      1. Полученый массив объектов кладёт в выходной параметр ПодходящийФормальныйЭлемент. Возвращает управление выше.
      2. Если массив объектов пуст, выкидывает исключение: «Я не знаю, что такое <слово>».

## Алгоритм «Сопоставление синонима с началом фразы»

Алгоритм проверяет, является ли Синоним (строка) началом Фразы (массив строк).

### Входные параметры

* Синоним – Текст
* Фраза – массив строк

### Выходные параметры

* Кол-во слов, использованных из Фразы

### Алгоритм

1. Разбивает Синоним на массив слов по разделителям (аналогично Фразе)
2. Если синоним длиннее фразы, выкидывает исключение «Я бы понял, если бы вы упомянули “<синоним>”»
3. Перебирает слова Синонима, сравнивает с соответсвующим словом начала Фразы.
   1. Если все слова совпали, возвращает кол-во слов
   2. Иначе выкидывает исключение «Не знаю, что такое “<первое неподошедшее слово Фразы>”»

## Алгоритм «Сопоставления слова фразы с объектом»

### Входные параметры

* Падеж
* Слово

### Выходные параметры

* Объекты – массив Объектов

### Алгоритм

1. Проходит по всем объектам (предметам, потом локациям), сравнивая Слово с падежной формой объекта в Падеже в словаре
   1. Все подходящие складывает в исходящий параметр

## Алгоритм «Очистка списка ПодходящихШаблонов от недоступных объектов»

### Входные параметры

* ПодходящиеШаблоны

### Выходные параметры

* ПодходящиеШаблоны

### Алгоритм

1. Перебирает все ПодходящиеШаблоны, проверяя все объекты на доступность (Локация должна быть текущей, Предметы должны быть либо в текущей локации, либо в инвентаре игрока)
   1. Сохраняет первый недоступный объект в переменной
   2. Если в ПодходящемШаблоне все объекты доступны, добавляет его в исходящий параметр
2. Если после проверки в исходящем параметре нет ни одной записи, выбрасывает исключение «Я не вижу здесь <сохранённый недоступный объект>»