Защищено:
Гапанюк Ю.Е.

"__"_____**2023** г.

Демонстрация:
Казакова В.В.

"__"_____**2023** г.

Отчет по домашней работе по курсу Парадигмы и конструкции языков программирования

Тема работы: " Ранее неизвестный язык программирования. "

10
(количество листов)

ИСПОЛНИТЕЛЬ:

студент группы ИУ5Ц-51Б _____
(подпись)

Казакова В.В. "__"_____**2023** г.

Москва, МГТУ - **2023**

Содержание

1. Описание задания

1. Выберите язык программирования (который Вы ранее не изучали) и (1) напишите по нему реферат с примерами кода или (2) реализуйте на нем небольшой проект (с детальным текстовым описанием).

2. Реферат (проект) может быть посвящен отдельному аспекту (аспектам) языка или содержать решение какой-либо задачи на этом языке.

3. Необходимо установить на свой компьютер компилятор (интерпретатор, транспилятор) этого языка и произвольную среду разработки.

4. В случае написания реферата необходимо разработать и откомпилировать примеры кода (или модифицировать стандартные примеры).

5. В случае создания проекта необходимо детально комментировать код.

6. При написании реферата (создании проекта) необходимо изучить и корректно использовать особенности парадигмы языка и основных конструкций данного языка.

7. Приветствуется написание черновика статьи по результатам выполнения ДЗ. Черновик статьи может быть подготовлен группой студентов, которые исследовали один и тот же аспект в нескольких языках или решили одинаковую задачу на нескольких языках.

2. Описание выбранного языка программирования

Язык программирования "English" является уникальным и инновационным инструментом разработки, который отличается от других существующих языков программирования. Благодаря своему уникальному подходу и возможностям, он предоставляет разработчикам полное преимущество и гибкость при создании программного обеспечения.

Язык программирования "English" был разработан в 1972 году компанией Interlingua Corporation. Этот язык был создан с целью предоставить программистам удобный и эффективный инструмент для разработки ПО полностью на английском языке.

Одной из главных особенностей "English" является его простота и лаконичность. Язык разработан таким образом, чтобы облегчить понимание и использование для разработчиков всех уровней. Он избегает излишней сложности и лишних синтаксических правил, позволяя программистам фокусироваться на самом процессе создания программы.

Также стоит отметить мощные возможности "English" в области связывания и интеграции. Язык имеет широкий набор инструментов и библиотек, которые позволяют разработчикам эффективно взаимодействовать с другими приложениями и системами. Благодаря этому, создание сложных программных решений становится более эффективным и удобным.

Еще одним важным аспектом "English" является возможность использования естественного языка в кодировании. Разработчики могут использовать английский язык для написания комментариев, описания функций и переменных, что делает код более понятным и доступным для других программистов. Это особенно полезно для командной разработки, где обмен информацией играет важную роль.

Необходимо также отметить, что "English" подходит для разработки различных видов приложений, включая веб-приложения, мобильные приложения, игры, настольные программы и многое другое. Он предлагает

мощные инструменты для работы с сетевыми протоколами, базами данных и взаимодействия с аппаратными компонентами устройств.

Благодаря своей природе "English" является языком программирования, который продолжает развиваться и обновляться. Разработчики всегда стремятся улучшить его функциональность, расширить возможности и улучшить производительность.

В итоге, язык программирования "English" представляет собой захватывающий инструмент, который вдохновляет профессионалов разработки на создание инновационных и качественных программных решений. Его простота, гибкость и интеграционные возможности делают его незаменимым инструментом в современной разработке программного обеспечения.

3. Программы, написанные на ЯП ~English и экранные формы

~English (NOT English) – это эзотерическая попытка использовать естественный язык в качестве языка программирования.

Оператор – это предложение на английском языке. Синтаксис языка очень свободный, поэтому можно писать бессмысленные предложения. Ключевые слова всегда имеют несколько синонимов.

Вот 7 типов утверждений в ~English:

Выражение	Значение
Stop	Останавливает программу. Все, что после "Stop" игнорируется, "Остановитесь и сделайте что-нибудь еще". Таким образом, синтаксически допустимо.
While	Обеспечивает функциональность цикла: цикл выполняется до тех пор, пока выполняется указанное условие.
Display	Отображает число или строку.
Declaration	Объявляет новую переменную с универсальным типом.
If ... Else	Условное утверждение.
Function call	Вызывает функцию.

Пользователь может не определять функции, в язык встроен набор вызываемых функций. Этими функциями являются:

- to number (пронумеровать, посчитать)
- to string (вывести строку)
- get input (получить входные данные)
- ask (получить выходные данные)

~English revised – это слегка измененная версия оригинального языка программирования ~English. В нем предпринята попытка обобщить некоторые языковые возможности и добавлено значительное количество новых. В частности, он предоставляет следующие дополнительные функциональные возможности:

- Введен новый синтаксис вызова функций. Это позволяет сделать функцию "отображаемой". Старый синтаксис вызова функции больше не поддерживается, поскольку он выглядел не очень естественно.

- Программы могут содержать комментарии (notes) в дополнение к коду.
- Часто используемые значения могут быть представлены некоторыми встроенными константами. Например, можно написать "один" вместо 1.
- "It" (и "its") может использоваться для ссылки на последнюю использованную переменную.
- Возможно создание пользовательских функций. Эти функции принимают свои аргументы по ссылке. Также был введен оператор return.
- Предложения могут заканчиваться на "!" или "?".
- Переменные были обобщены до объектов. То, что раньше было переменной, теперь является объектом типа "переменная". Объекты могут иметь данные-члены, и над ними могут быть определены операции. Для этого используется утиная типизация (хотя тип переменной является универсальным типом).
- Библиотеки могут быть написаны как общие объекты или модули python. Пользователи могут создавать свои собственные библиотеки. Библиотеки, созданные пользователем (с использованием C/C++)
- Существует стандартная библиотека, которая добавляет поддержку массивов, строк и сокетов. Также будут интегрированы другие библиотеки, такие как математические или системные библиотеки операционной системы.
- Эта пересмотренная версия явно не полностью совместима с оригинальной версией.

На Github доступен интерпретатор C++ (основанный на исходном коде оригинального интерпретатора), но он еще не полностью реализован.

Компилятором ~English (обычно для какого-либо другого языка высокого уровня) обычно является программист. Обычно это люди, и иногда они могут быть довольно глючными. Однако программисты не в состоянии скомпилировать ~English на другой язык высокого уровня, если исходный код

невычислим. Компиляторы для программистов занимают больше времени, чем другие компиляторы, такие как "GCC" или около того, но они также обладают способностью дизассемблировать или декомпилировать язык высокого уровня ~English.

```
This program writes "Hello World" (without quotes) to the output.
```

```
This program outputs Hello World.
```

Использование Openais JavaScript beta sandbox. Исходный код на ~English написан в виде комментария над кодом.

```
/* This program outputs Hello World. */
var helloWorld = document.createElement('div');
helloWorld.innerHTML = 'Hello World';
document.body.appendChild(helloWorld);
```

Как видно из примера с Hello World, существует несколько способов выразить одну и ту же идею. И из-за неоднозначной природы человеческих языков то же самое верно и для интерпретации программ. Следовательно, в ~English могут быть разные допустимые интерпретации одной и той же программы.

Некоторые компьютерные программы также способны компилировать ~English. Они называются системами программирования на естественном языке. Однако в данный момент не существует отдельных компилятора и сред разработки, в который можно писать код на языке программирования ~English, так как язык достаточно новый.

Ниже представлены примеры других программ, написанных на ~English:

```
This program outputs 99 verses in this manner:
"N bottles of beer on the wall,
N bottles of beer.
Take one down, pass it around,
N-1 bottles of beer on the wall",
where N is replaced by numbers from 99 to 0
and 1-1=0 is replaced by "No more".
If this is the 100th verse,
replace the second-to-last line by:
"Go to the store and buy some more"
and replace 0-1 by 99.
```

Использование Openais JavaScript beta sandbox. Исходный код на ~English написан в виде комментария над кодом.

```
/* This program outputs 99 verses in this manner: "N bottles of beer on the wall, N bottles of beer. Take one down, pass it around, N-1 bottles of beer on the wall", where N is replaced by numbers from 99 to 0 
var bottles = 99;
while (bottles >= 0) {
  var verse = document.createElement('div');
  verse.innerHTML = bottles + ' bottles of beer on the wall, ' + bottles + ' bottles of beer. Take one down, pass it around, ' + (bottles - 1) + ' bottles of beer on the wall';
  if (bottles == 0) {
    verse.innerHTML = 'No more bottles of beer on the wall, no more bottles of beer. Go to the store and buy some more, 99 bottles of beer on the wall';
  }
  document.body.appendChild(verse);
  bottles--;
}
```

```
99 to 0 and 1-1=0 is replaced by "No more". If this is the 100th verse, replace the second-to-last line by: "Go to the store and buy some more" and replace 0-1 by 99. */
```

```
Create a variable named name.
Display "Enter your name: " and a newline.
Vary the value of name by call the function "ask".
Make another variable called age.
Display "Enter your age: " and a newline.
Set the value of age to call function "to number" with call another function "ask".
While the age is smaller than 0 do:
    If the age equals -100 then:
        Display "That seems highly unlikely..." and a newline.
    That's it.
    Display "Enter a valid age, " and the name.
    Display ":" and a newline.
    Set the value of age to call function "to number" with call another function "ask".
That's all.
Display "Your name is ", the name, "." and a newline.
Show "Your age is ", the age, "." and another newline.
If the name equals "World" then:
    Display "Hello World!" and a newline.
    Display another newline.
That's all.
Else, then:
    Display "Now this is no longer a 'Hello, world!' program..." and a newline.
That's all.
If the age is larger than 100 or the age equals 100 then:
    Display "You are very old." and a newline.
    Display another newline.
That's all.
Otherwise, then:
    Display "You are less than 100 years old." and a newline.
That's it.
Show "Bye!" and a newline.
Stop this program.
```

```
This program tosses a coin and outputs "Yes" if it's tails or "No" if it's heads.
```