**Введение**

В приложении для Excel было решено использовать язык программирования Go.

Немного об истории.

Go был создан в лабораториях Google и появился в 2009 году. Его автором является Роб Пайк,

один из создателей ОС UNIX и Plan9. Go поддерживает следующие платформы:

Windows, MacOS, Linux, BSD и другие. Существуют версии для разных процессорных архитектур, в том числе под x86 и x64.

Почему именно Go?

Go – открытый компилируемый язык со статической типизацией. По словам создателей, Go лишен следующих недостатков:

* неконтролируемые зависимости;
* использование разными программистами разных подмножеств языка;
* затруднения с пониманием программ, вызванные неудобочитаемостью кода, плохим документированием и так далее;
* дублирование разработок;
* высокую стоимость обновлений;
* несинхронные обновления при дублировании кода;
* сложность разработки инструментария;
* проблемы межъязыкового взаимодействия.

Основными требованиями создателей к языку стали:

* Ортогональность. Язык должен предоставлять небольшое число средств, не повторяющих функциональность друг друга.
* Простая и регулярная грамматика. Минимум ключевых слов, простая, легко разбираемая грамматическая структура, легко читаемый код.
* Простая работа с типами. Типизация должна обеспечивать безопасность, но не превращаться в бюрократию, лишь увеличивающую код. Отказ от иерархии типов, но с сохранением объектно-ориентированных возможностей.
* Отсутствие неявных преобразований.
* Сборка мусора.
* Встроенные средства распараллеливания, простые и эффективные.
* Поддержка строк, ассоциативных массивов и коммуникационных каналов.
* Чёткое разделение интерфейса и реализации.
* Эффективная система пакетов с явным указанием зависимостей, обеспечивающая быструю сборку.

Благодаря своей компилируемой природе, Go работает очень быстро. Это выгодно отличает его от подобных ему Javascript NodeJS и Python.

Есть другие компилируемые языки, которые работают так же быстро как Go.

Но был выбран именно он потому что:

1. У Go огромное сообщество.

В случае возникновения проблем всегда найдутся люди, которые могут помочь решить их.

2. Создатель Go – гигантская корпорация.

За Go стоит Google, а это значит стабильные и обратно совместимые обновления гарантированы.

3. Огромное кол-во готовых библиотек

Сообщество поддерживает огромное количество библиотек.

Для большинства задач уже есть готовое и высококачественное решение.

Исходя из этого, можно заявить, что язык целиком подходит под предъявляемые требования.

**Описание программного кода**

Работа программы была протестирована на операционных системах Linux и Windows.

package main

import (

"fmt"

"github.com/360EntSecGroup-Skylar/excelize"

"strconv"

)

func indexOf(val string,arr [8]string)int{ //возвращает индекс элемента в массиве

for i,v:=range arr{

if val==v{

return i;

}

}

return -1;

}

func main(){

f, \_ := excelize.OpenFile("lib.xlsx") //открываем таблицу

var books [8] string; //названия книг

var values [8] int64; //кол-во книг

for i:=0; i<8; i++{

cell:=fmt.Sprintf("A%d",i+3)

books[i],\_=f.GetCellValue("Лист3", cell)

cell=fmt.Sprintf("B%d",i+3)

tmp,\_:=f.GetCellValue("Лист3",cell);

values[i],\_=strconv.ParseInt(tmp,10,64) //covert string to int64

}

rows,\_:=f.GetRows("Лист4")

for i:=0;i<len(rows[15]);i++{

if i>29{

index:=indexOf(rows[15][1+2\*i],books)

if index!=-1{

values[index]--;

}

break

}

index:=indexOf(rows[15][1+3\*i],books)

if index!=-1{

values[index]--;

index:=indexOf(rows[16][1+3\*i],books)

if index!=-1{

values[index]--;

}

}

}

for i:=0; i<len(values); i++{

cell:=fmt.Sprintf("B%d",i+3)

f.SetCellValue("Лист4",cell,values[i])

}

f.SaveAs("lib.xlsx")

}

Go – язык, в котором любая программа является пакетом.

В первом строчке мы объявляем пакет main.

Далее идет секция импортирования сторонних модулей.

Модули fmt и strconv – встроенные, “github.com/360EntSecGroup-Skylar/excelize” - сторонний, предварительно скаченный модуль. Далее объявляется функция indexOf для поиска вхождения элемента в массив. Затем объявляется функция main.

В языке Go main – точка входа в программу.

Т.е эта функция вызывается сразу после начала выполнения программы и является главной.

Из нее вызываются все остальные функции. Переходим к рассмотрению тела функции main.

Первом делом файл открывается lib.xlsx – таблица с данными. Далее объвляются переменные books и values для хранения списка книг и кол-ва книг соответственно.

Далее в цикле заполняются эти массивы. С помощью функции GetCellValue получается значение ячейки, функция Sprintf позволяем “склеить” строку и число для получения адреса ячейки. И функция ParseInt из модуля strconv конвертирует строку из ячейки в число.

В переменную \_ записывается значение ошибки, если она возникает.

Но так как мы используем готовую таблицу, ошибка возникнуть не может, поэтому переменная не используется.

Следующим шагом требуется получить содержание другого,4 листа таблицы с помощью команды f.GetRows, возвращающей двумерный массив (массив, элементами которого являются массивы). В цикле перебираются элементы 16 строки, начиная с начала.

index:=indexOf(rows[15][1+3\*i],books)

Эта строчка получает индекс текущей книги, если она есть.

Т.е функция возвращает -1 если ячейки таблица не книга, и ее номер если это книга.

if index!=-1{

values[index]--;

}

Если нашлась книга, то уменьшаем на одну книгу в библиотеке (уменьшается элемент массива, который потом записывается в соответствующие ячейки таблицы)

Ниже происходит проверка есть ли 2 книга у данного человека, а на несклько строк выше код, который вовремя останавливает цикл, чтобы не выйти за границы массива.

Ну и наконец заключительный цикл, который изменяет значение кол-ва книг из элементов массива values и командой f.SaveAs(“lib.xlsx”) файл сохраняется.