Міністерство освіти, науки, молоді та спорту України

Львівський національний університет імені Івана Франка

Кафедра фізики напівпровідників

Самостійна робота

« Вивчення офісного пакету Google Docs.

Взаємодія з Google Spread Sheets за допомогою мови PYTHON»

Студентки

Групи ФеІ – 22

Литвин Віри

Перевірив

доц. Павлишенко Б.М.

Львів 2012

**Google Docs**

## 1. Google Docs

**Google Docs** (укр. *Документи Ґуґл*) - розроблений Google безкоштовний мережевий офісний пакет, що включає текстовий, табличний редактор, службу для створення презентацій та багато іншого.

Утворений у результаті злиття Writely і Google Spreadsheets.

Це веб-орієнтована програма, що працює в рамках веб-браузера без установлення на комп'ютер користувача. Документи і таблиці, що створюються користувачем, зберігаються на сервері Google, або можуть бути збережені у файл. Це одна з ключових переваг програми, оскільки доступ до введених даних може здійснюватися з будь-якого комп’ютера, під’єднаного до інтернету. Доступ до особистих документів захищений паролем.

**Переваги :**

## Доступ до роботи будь-де і будь-коли

Документи Google надійно працюють на основі веб-технологій, даючи змогу ефективно виконувати ділові задачі на робочому місці, у дорозі, удома та з мобільного телефону навіть у режимі офлайн.

## Сумісність з операційними системами

Документи Google працюють у веб-переглядачі на ПК, комп'ютерах Mac і Linux і підтримують такі популярні формати: .doc, .xls, .ppt і .pdf.

## Легке завантаження файлів і надання до них спільного доступу

Файли, які зберігаються в Документах Google, завжди доступні та мають резервну копію онлайн.

## Безпечні елементи керування доступом

Адміністратори можуть керувати правами доступу до файлів у межах системи, а власники документів мають змогу надавати та скасовувати доступ до файлів у будь-який час.

### 1.[1. Writely](http://uk.wikipedia.org/wiki/Google_Docs#Writely)

Це текстовий процесор, що дозволяє редагувати текстові документи OpenDocument, Microsoft Word, а також електронні таблиці. Створений за допомогою технології AJAX.

Доступна велика кількість засобів форматування: зміна розміру і стилю шрифту, вибір кольору та оздоблення, створення списків і таблиць, вставка картинок, посилань і спеціальних знаків. Можна робити закладки, коментарі.

Зберігаються документи автоматично, з мірою внесення змін, але кожна правка записується, і можна користуватися функцією скасування і повернення змін так само, як і у звичайному текстовому редакторі. Існує можливість завантажувати на сервер і скачувати з нього файли у різних форматах.

Підтримується простий текст, HTML, Microsoft Word, RTF, OpenDocument, PDF і кілька графічних форматів. Можна отримати підбірку текстів у вигляді файлів HTML в архіві ZIP.

### 1.2. [Google Spreadsheets](http://uk.wikipedia.org/wiki/Google_Docs#Google_Spreadsheets)

Дозволяє заносити дані в рядки і стовпчики електронної таблиці, а також проводити нескладні обчислення. Програма допускає перетворення даних і у формат Microsoft Excel, і в загальнодоступні OpenDocument і CSV.

Передбачена можливість доступу інших користувачів до створеної таблиці, як з правом редагування, так і без.

Це програмне забезпечення має трошки менше можливостей, порівняно з повноцінними електронними таблицями, такими, як Microsoft Excel або OpenOffice.org Calc , але цілком достатньо для простих дій з даними і воно постійно вдосконалюється, набуваючи все нових і нових функцій.

### 1.3. Google Presentations

Дає можливість створювати електронні презентації. Реалізована можливість імпорту/експорту файлів Microsoft PowerPoint (\*.ppt).

У основу Google Presentations лягли розробки компанії Tonic System, придбаної компанією Google в 2007 році.

## Передача права власності на Документи Google

Примітка. Ця функція доступна лише в облікових записах, які перейшли нанову інфраструктуру облікових записів Google Apps.

Існує три типи користувачів у Документах Google: власник, редактор і користувач із правами перегляду. Одночасно в документів може бути багато редакторів і користувачів із правами перегляду, але тільки один власник. За умовчанням автор документа є також його власником, але право власності можна передати іншому користувачу.

**Якщо ви є суперадміністратором** Google Apps, ви можете передавати право власності на всі документи від одного користувача іншому у своїй організації. Не можна передавати документи, власником яких є видалений користувач. Після передачі первинний власник зберігає за собою право редагування документів, якщо цього користувача або його дозволи на редагування не видалено.

Передача права власності на всі документи користувача може бути корисною, якщо користувач залишає організацію. Під час видалення користувача з домену вам буде запропоновано передати право власності на його документи іншому користувачу, щоб не втратити їх.

# Мова PYTHON

## 2.1. Загальний огляд мови

**Python** (рекомендоване прочитання - « **Па́йтон** », запозичено назвуз британського шоу Монті Python) - інтерпретована об'єктно-орієнтована мова програмування високого рівня з динамічною семантикою.

Розроблена в 1990 році Гвідо ван Россумом.

Структури даних високого рівня разом із динамічною семантикою та динамічним зв'язуванням роблять її привабливою для швидкої розробки програм, а також як засіб поєднання існуючих компонент.

Python підтримує модулі та пакети модулів, що сприяє модульності та повторному використанню коду.

Інтерпретатор Python та стандартні бібліотеки доступні як у скомпільованій так і у вихідній формі на всіх основних платформах.

В мові програмування Python підтримується декілька парадигм програмування, зокрема: об'єктно-орієнтована, процедурна, функціональна та аспектно-орієнтована.

Серед основних переваг Python можна виділити такі:

* чистий синтаксис (для виділення блоків слід використовувати пробіли);
* портабельність (що властиве більшості інтерпретованих мов);
* стандартний дистрибутив має велику кількість корисних модулів (включно з модулем для розробки графічного інтерфейсу);
* можливість використання Python в діалоговому режимі (дуже корисне для експериментування та вирішення простеньких задач);
* стандартний дистрибутив має простеньке, але разом досить потужне середовище розробки, яке зветься IDLE і яке написане мовою Python;
* зручний для вирішення математичних проблем (розуміє комплексні числа, може оперувати з цілими числами довільної величини, у діалоговому режимі може використовуватися як потужний калькулятор).

Його легко використовувати, у той же час це справжня мова програмування, що має значно багатшу структуру та підтримку для написання великих програм, ніж традиційна оболонка.

З іншого боку, Python пропонує набагато кращу перевірку помилок ніж С, і, будучи мовою *дуже високого рівня*, він має вмонтовані типи даних високого рівня, зокрема гнучкі масиви та словники, для ефективного створення яких на мові С потрібно кілька днів.

Завдяки загальнішім типам даних, Python може бути застосований для вирішення набагато ширшого кола проблем, ніж така мова як AWK, чи навіть Perl; водночас багато речей можуть бути створені на Python так само просто, як і на цих мовах.

Python дозволяє поділити нашу програму на модулі, які можуть згодом використовуватися в інших програмах, написаних на Python. Python має велику кількість стандартних модулів, які ви можете покласти в основу своїх програм або на яких можете почати вчитися програмувати. Існують також вбудовані модулі для таких речей як файловий ввід-вивід, системні виклики, сокети, і навіть графічні інтерфейси.

Python - це інтерпретована мова, що може зберегти вам чимало часу при розробці програм, тому що компіляція та зв'язування (linking) не потрібні. Інтерпретатор може також використовуватися у діалоговому режимі, що спрощує експериментування з різними рисами мови, написання одноразових програм чи тестування шляхом зворотної розробки. Це також зручний настільний калькулятор.

Python дозволяє створювати дуже компактні та читабельні програми. Типова програма, написана на Python - набагато коротша ніж еквівалентна програма на C чи C++, що відбувається з таких причин:

* структури даних високого рівня дозволяють виразити складні операції за допомогою окремих тверджень;
* групування тверджень робиться за допомогою виділення пробілами замість фігурних дужок;
* декларація змінних чи аргументів непотрібна;

Python *можна розширити*: якщо ви вмієте програмувати на C, то вам буде легко додати нову вбудовану функцію до інтерпретатора, що надасть можливість виконувати критичні операції з максимальною швидкістю, або прив'язати Python до вже скомпільованих бібліотек (напр., спеціальні графічні бібліотеки). Також ви можете прив'язати інтерпретатор Python до програми, написаної на C і використовувати його як розширення чи командну мову для вашої програми.

На основі Python було створено кілька спеціалізованих підмножин мови, в основному призначених для статичної компіляції в машинний код. Деякі з них:

* RPython - створена в рамках проекту PyPy значно обмежена реалізація Python без динамізму часу виконання та деяких інших можливостей. RPython код можна компілювати в безліч інших мов/платформ - C, JavaScript, Lisp, .NET, LLVM. На RPython написаний інтерпретатор PyPy.
* Pyrex - обмежена реалізація Python, але трохи менше, ніж RPython. PyReX розширено можливостями статичної типізації типами з мови С і він дозволяє вільно змішувати типізований та не типізований код. Призначений для написання модулів розширень, компілюється в код на мові С.
* Cython - розширена версія Pyrex.
* Pyastra - компілятор Python коду в асемблер для PIC архітектури.
* shed-skin - призначений для компіляції неявно статично типізованого Python коду в оптимізований код на мові С++, проект далекий від завершення.

# Практична частина

## 1. Інтерпретація результатів.

Для демонстрації можливостей мови Python написано програму, за допомогою якої можна відправити певні дані в Google Spread Sheets не вводячи їх безпосередньо у таблицю, а використовуючи зручне віконечко відправляти одним натиском мишки на відповідній кнопці цілу групу інформації.

Для цього заздалегідь створено документ Google Spread Sheets, і контролюється його редагування одною особою - власником ,але переглядати та редагувати таблицю можуть усі, хто отримає на неї посилання.

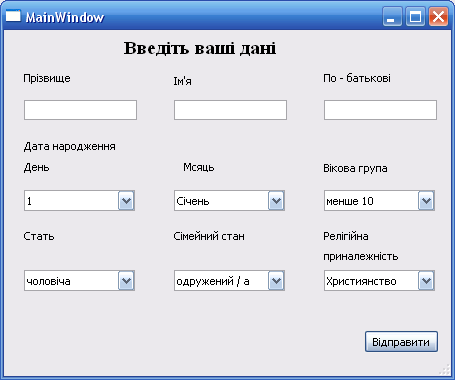
* Документи в Google Docs можуть створювати лише власники акаунтів у Google.

## 2. Розробка програми

## 2.1. Проектування інтерфейсу.

Для створення графічного інтерфейсу використано бібліотеки Qt, а саме обрано PyQt4. Оболонка, як на мене, має дуже зручний інтерфейс і багато корисних та цікавих можливостей.

Використання такого середовище дозволяє заощадити багато часу, який зайняло б програмування інтерфейсу вручну.

Отож в Qt Designer створено таке вікно :

### 2.2. Компілювання програми.

Щоб цей інтерфейс можна було використовувати в моїй програмі його слід конвертувати у файл з розширенням \*.py .

Для цього потрібно відкрити командний рядок і написати там наступний текст : **pyuic4 form1.ui > editor.py,**

де : **pyuic4** глобальна змінна системи, замість неї можна прописати повний шлях до цього файлу; **form1.ui** - ім’я файлу, що конвертується, а **editor.py** – ім’я файлу, в який конвертується.

Тоді в середовищі ***Python IDLE*** написано код програми, використовуючи спеціальні бібліотеки для доступу до Spread Sheets на мові Python, що містяться в додатковому модулі gdata-2.0.15, який можна безкоштовно завантажити з сайту Google.

Файл коду має розширення \*.py.

Проте, щоб скомпілювати програму і створити виконуваний файл потрібно проробити ще кілька операцій в командному рядку. Складна компільованість вважається головним недоліком цієї мови.

Отож, для створення \*.exe спочатку потрібно створити \*.spec . Для цього прописується в командній стрічці наступне:

* *<шлях до програми >\pyinstaller-1.5.1 >python Makespec.py --onefile -w <шлях до скрипту >\\*.py*

В результаті успішного виконання цієї команди утвориться файл \*.spec і папка для проекту всередині директорії <шлях до програми >\pyinstaller-1.5.1.

Далі наступна команда для створення безпосередньо файлу \*.exe :

* <шлях до програми >\pyinstaller-1.5.1 >python Build.py \*\\*.spec

В останніх двох командах python - глобальна змінна, вказівник на файл python.exe, замість цього можна вказати повний шлях до файлу.

Після запуску отриманого файлу відкривається вікно, зображене вище. Користувач введе у відповідні комірки прізвище ім’я та по-батькові, виберемо дату народження, вік стать, сімейний стан та релігійну приналежність. Далі натискає кнопку «Відправити».

Перейшовши за наступним посиланням користувач побачить документ Google Spread Sheets – табличку, в якій відображені введені ним дані.

* [*https://docs.google.com/spreadsheet/ccc?key=0AsFKUoTWv9sgdC01dFR1WVZ4RjVlc2JFbmY5VXVIWEE*](https://docs.google.com/spreadsheet/ccc?key=0AsFKUoTWv9sgdC01dFR1WVZ4RjVlc2JFbmY5VXVIWEE)

Для коректної роботи програми необхідно стабільне підключення до інтернету, оскільки її робота базується на з’єднанні з серверами Google.

**Висновки**

**Python** - це потужна мова програмування, якою легко оволодіти. Вона має ефективні структури даних високого рівня та простий, але ефективний підхід до об'єктно-орієнтованого програмування.

Елегантний синтаксис Python, динамічна обробка типів, а також те, що це інтерпретована мова, роблять його ідеальним для написання скриптів та швидкої розробки прикладних програм у багатьох галузях на більшості платформ.

Є широкий асортимент безкоштовних середовищ, модулів, бібліотек та графічних оболонок для роботи з Python, тому кожен користувач неодмінно знайде ідеальну для себе.

Інтерпретатор мови Python і багата стандартна бібліотека (як код-джерело, так і бінарні дистрибутиви для усіх головних операційних систем) можуть бути отримані з сайту Python, і можуть вільно розповсюджуватися. Цей же сайт має дистрибутиви та посилання на численні модулі, програми, утиліти та додаткову документацію.

Завдяки портованості код на Python легко поєднувати з програмами, які написані іншими мовами.

Інтерпретатор мови Python може бути легко розширений функціями та типами даних, розробленими на C чи C++ (або на іншій мові, яку можна викликати із C). Python також зручний як мова розширення для прикладних програм, що потребують подальшого налагодження.

**Джерела:**

1. Guido van Rossum, Fred L. Drake, Jr. «Python Tutorial»
2. http://www.python.org/
3. <http://younglinux.info/oopython.php> http://code.google.com/intl/uk-UA/apis/spreadsheets/data/1.0/developers\_guide\_python.html
4. http://uk.wikipedia.org/
5. http://manuals.u-medio.com/index.php?title=%D0%9E%D1%81%D0%BD%D0%BE%D0%B2%D0%B8\_Python\_%D0%B2\_%D0%BA%D0%BE%D1%80%D0%BE%D1%82%D0%BA%D0%BE%D0%BC%D1%83\_%D0%B2%D0%B8%D0%BA%D0%BB%D0%B0%D0%B4%D1%96.\_%D0%93%D0%BB%D0%B0%D0%B2%D0%B0\_1

**Код програми :**

# code.py

# Virusya Lytvyn

# -\*- coding: utf-8 -\*-

import sys

import gdata.spreadsheet.service

import gdata.service

import atom.service

import gdata.spreadsheet

import atom

import gdata

from PyQt4 import QtCore, QtGui

from editor import Ui\_MainWindow

class GoogleSpreadSheet(QtGui.QMainWindow):

def \_\_init\_\_(self, parent=None):

QtGui.QWidget.\_\_init\_\_(self, parent)

self.ui = Ui\_MainWindow()

self.ui.setupUi(self)

self.connect(self.ui.sendButton, QtCore.SIGNAL("clicked()"), self.readDataFromForm)

def readDataFromForm(self):

#print "button %s was pressed" % self.sender()

firstName = unicode(self.ui.firstnameLineEdit.text())

secondName = unicode(self.ui.secondnameLineEdit.text())

lastName = unicode(self.ui.lastnameLineEdit.text())

birthday = unicode(self.ui.CBday.itemText(self.ui.CBday.currentIndex()))

month = unicode(self.ui.CBmounce.itemText(self.ui.CBmounce.currentIndex()))

year = unicode(self.ui.CByear.itemText(self.ui.CByear.currentIndex()))

malefemale = unicode(self.ui.CBmale.itemText(self.ui.CBmale.currentIndex()))

married = unicode(self.ui.CBmerr.itemText(self.ui.CBmerr.currentIndex()))

religious = unicode(self.ui.CBChist.itemText(self.ui.CBChist.currentIndex()))

## firstName = unicode(firstName)

## secondName = unicode(secondName)

## print married

## print religious

gd\_client = gdata.spreadsheet.service.SpreadsheetsService()

gd\_client.email = 'baterfluy@gmail.com'

gd\_client.password = '369741369'

gd\_client.ProgrammaticLogin()

spreadsheet\_feed = gd\_client.GetSpreadsheetsFeed()

for spreadsheet in spreadsheet\_feed.entry:

print spreadsheet.id.text.rsplit('/', 1)[1]

spreadsheet\_key = spreadsheet.id.text.rsplit('/', 1)[1]

feed = gd\_client.GetWorksheetsFeed(spreadsheet\_key)

for worksheet in feed.entry:

worksheet\_key = worksheet.id.text.rsplit('/', 1)[1]

print worksheet\_key

gd\_client = gdata.spreadsheet.service.SpreadsheetsService(spreadsheet\_key, worksheet\_key)

gd\_client.email = 'baterfluy@gmail.com'

gd\_client.password = '369741369'

gd\_client.ProgrammaticLogin()

dict = {'firstname':firstName, 'secondname':secondName, 'lastname':lastName,

'birthdayday':birthday, 'birthdaymonth':month, 'age':year,

'malefemale':malefemale, 'marriage':married, 'religion':religious}

#dict = {'month':month}

#gd\_client.InsertRow(dict, spreadsheet\_key, worksheet\_key)

entry = gd\_client.InsertRow(dict, spreadsheet\_key, worksheet\_key)

if isinstance(entry, gdata.spreadsheet.SpreadsheetsList):

print "Insert row is success."

#Run the program

if \_\_name\_\_ == "\_\_main\_\_":

app = QtGui.QApplication(sys.argv)

myapp = GoogleSpreadSheet()

myapp.show()

sys.exit(app.exec\_())