le Générateur et Validateur d'IBAN

Limites et points à améliorer: X
Points forts du programme: V

Deux scripts principaux composent l'application :

- main.py: gère la lecture des données, la génération de l'IBAN et les calculs de clé.
- **IBANGeneratorGUI.py**: fournit l'interface graphique utilisateur (GUI) avec Tkinter et la validation via API

Le fichier texte attendu a un format simple, par exemple :

```
Code banque 20041
Code agence 01005
Numero de compte bancaire 0500013M026
Chiffre d'indicatif national 06
```

La fonction lire_infos() extrait uniquement la dernière valeur de chaque ligne (après découpage par espaces/tabulations).

Dépendance à un format strict du fichier texte : si une ligne est mal formée, le programme échoue.

Le cœur du calcul se fait dans calculer cle iban() :

- 1. Construction d'un IBAN provisoire avec une clé fictive 00.
- 2. Déplacement du code pays + clé à la fin de la chaîne.
- 3. Conversion des lettres en chiffres (A=10, B=11, ..., Z=35).
- 4. Application de l'algorithme de modulo 97.
- 5. Calcul de la clé de contrôle (98 reste).

Algorithme conforme : le calcul de la clé de contrôle suit la norme officielle (ISO 7064 mod97-10).

Exemple de sortie dans le terminal :

```
IBAN provisoire : FR0020041010050500013M02606
IBAN rearrange : 20041010050500013M02606FR00
IBAN numerique : 200410100505000132602615152700
Reste final : 84
Cle calculee : 14
```

Le résultat final est un IBAN complet : FR1420041010050500013M02606

X Pas de gestion des autres pays : le programme est limité aux IBAN français (préfixe FR).

L'interface permet :

- Charger un fichier .txt via un bouton.
- Afficher les informations bancaires extraites.
- Générer automatiquement l'IBAN.
- Valider l'IBAN en appelant l'API publique ibanapi.com.

X Interface rudimentaire : l'UI est basique, sans options avancées (édition manuelle, export en PDF/CSV).

L'utilisateur reçoit une boîte de dialogue indiquant si l'IBAN est valide ou non.

La fonction action_valider_iban() interroge l'API avec une requête GET :

X Clé API exposée en clair dans le code : cela pose un problème de sécurité.

https://api.ibanapi.com/v1/validate/<IBAN>?api_key=<clé>

- Si la validation réussit (code 200), un message confirme que l'IBAN est valide.
- En cas d'échec (code 400), un message d'erreur s'affiche avec les raisons (ex. : mauvaise longueur, clé de contrôle incorrecte, etc.).
- Retour utilisateur clair : boîtes de dialogue pour erreurs, avertissements et succès.
- X Pas de retour détaillé sur la validité dans l'UI (ex. : différence entre une clé incorrecte et un mauvais format).

Les deux images jointes montrent la différence entre un IBAN valide (200) et un IBAN invalide (400)

```
StatusCode
                  : 200
StatusDescription : OK
Content
                      "result": 200,
                       "message": "Valid IBAN Number",
                       "validations": [
                          "result": 200,
                           "message": "Valid IBAN length"
                           "result": 200,
                          "message": "Valid IBAN C...
                  : HTTP/1.1 200 OK
RawContent
                    Transfer-Encoding: chunked
                    Connection: keep-alive
                    Nel: {"report_to":"cf-nel","success_fraction":0.0,"max_age":604800}
                    Vary: Accept-Encoding
                    X-Request-Id: HlHZpcalAeoPvphpwabUBZbbh...
Forms
Headers
                  : {[Transfer-Encoding, chunked], [Connection, keep-alive], [Nel, {"report_to":"cf-nel","success_fraction":0.0,"max_age":604800}], [Vary,
                    Accept-Encoding]...}
Images
InputFields
Links
ParsedHtml
                  : mshtml.HTMLDocumentClass
RawContentLength : 1050
PS C:\Users\User\OneDrive\Desktop\cours\Python> curl "https://api.ibanapi.com/v1/validate/FR4230066104640002073130156?api_key=261f79768b16828f01ba93b650631c
```

```
curl : { "result": 400, "message": "Invalid IBAN Number", "validations": [ { "result": 200, "message": "Valid IBAN length" }, { "result": 200, "message"
"Valid IBAN Checksum" }, { "result": 200, "message": "Valid IBAN Structure" }, { "result": 411, "message": "Invalid Account Number Checksum" } ],
"expremental": 0, "data": { "country_code": "FR", "iso_alpha3": "FRA", "country_name": "France", "currency_code": "EUR", "sepa_member": "Yes", "sepa_credit_transfer": "Yes", "sepa_credit_transfer": "Yes", "sepa_direct_debit": "Yes", "sepa_sdd_core": "No", "sepa_b2b": "Yes",
"sepa_card_clearing": "No" }, "bban": "30066104640002073130156", "bank_account": "00020731301", "bank": { "bank_name": "CREDIT INDUSTRIEL ET COMMERCIAL
CIC", "phone": "", "address": "57 R DE LA VICTOIRE", "bic": "CMCIFRPPCOR", "city": "PARIS 09 EME", "state": "", "zip": "75009" } } }
At line:1 char:1
                                                                                                                        InvalidOperation: (System.Net.HttpWebRequest:HttpWebRequest) [Invoke-WebRequest], WebException WebCmdletWebResponseException,Microsoft.PowerShell.Commands.InvokeWebRequestCommand
                     · CategoryInfo
· FullyQualifiedErrorId
```