

## 1 Objectifs

Le projet de la mineure « Python » permet de mettre en œuvre les principaux éléments du contenu du module dans le cadre de la conception d'une petite application en programmation objet. Le travail est à réaliser seul(e).

Rappel des modalités d'évaluation (MCC) : la mineure « Python » est en Contrôle Continu Intégral (CCI) : le projet compte pour la moitié de l'évaluation de ce module, l'autre moitié étant constituée par une moyenne pondérée des différents TP notés. Dans tous les cas, il n'y a pas de seconde session et les notes obtenues sont reportées en session 2.

## 2 Contexte du projet

L'objectif du projet est de créer une application permettant de manipuler des cartes de visites (virtual card) au format numérique et des événements d'agenda (rendez-vous, tâche à réaliser, etc.). Ce type de carte de visite est utilisé dans un contexte professionnel dans les carnets d'adresses des clients de messagerie ou en élément de la signature lors des échanges par mail par exemple, et ce type de calendrier d'événements est utilisé dans l'ENT pour vos emplois du temps par exemple.

Les principales actions de votre logiciel sont :

- la lecture et l'affichage (avec une mise en forme) d'un fichier de carte de visite (au format `vcf`) ou d'un fichier de calendrier (au format `ics`) en mode console et en mode graphique,
- la possibilité de modifier partiellement le contenu via une interface graphique et d'enregistrer ces modifications,
- la possibilité de transformer ce fichier en un fragment de code HTML exploitant les microformats<sup>1</sup> correspondants.

En mode console (terminal) « CLI » : les paramètres attendus sur la ligne de commande sont : en entrée, le nom du fichier (`vcf` / `ics`) à traiter et en sortie, le nom du fichier (`csv` / `html`) à produire. Si le fichier de sortie n'est pas spécifié, le programme doit afficher directement dans la console le contenu du fichier, ce contenu sera mis en forme. Si aucun paramètre n'est indiqué, le programme affiche les options possibles. Une des options permet d'afficher la liste des fichiers `vcf` et `ics` disponibles sur la machine en les recherchant, à partir du dossier spécifié, dans l'ensemble de ses sous-répertoires.

En mode graphique « GUI » : l'exploration d'une arborescence quelconque (et de ses éventuels dossiers) de fichiers permettant de lister uniquement les fichiers de type `vcf` et `ics` et leur emplacement est attendue. La sélection d'un de ces fichiers provoquera son affichage (mais on ne recherche pas nécessairement un affichage complet), la possibilité de modifier les principaux champs (là encore, on ne cherche pas à traiter tous les cas particuliers lors des modifications) et leur enregistrement (`csv` / `html`) et enfin l'option de génération du fragment HTML correspondant (seuls les éléments les plus significatifs sont à exporter). Il est donc possible de lancer une exécution en mode console, d'enregistrer le résultat puis de lancer une exécution en mode graphique et d'ouvrir directement la sauvegarde précédente (sans relire le fichier « `vcf` / `ics` » initial).

---

1. <https://fr.wikipedia.org/wiki/Microformat>

Quelques scénarii d'exécution (exemples fictifs de lancement de vos 2 programmes) :

```
python3 cli.py
python3 cli.py -h
python3 cli.py -d .
python3 cli.py -i edt.ics
python3 cli.py -i edt.ics -h fragment.html
python3 gui.py
```

Explications : les 5 premières commandes concernent le mode console (aussi appelé mode terminal ou fenêtre de commande) et la dernière commande permet de lancer l'interface graphique :

- la première ligne doit indiquer qu'il manque des paramètres et doit proposer de taper « `-h` » (ou « `--help` ») pour obtenir de l'aide ;
- la deuxième ligne affiche les modes d'utilisation de votre logiciel en mode console (i.e. les options possibles et leur rôle) ;
- la troisième ligne liste tous les fichiers `vcf` ou `ics` à partir du dossier spécifié (« `-d` » pour directory) [ici à partir du dossier courant (« `.` »)] en parcourant l'ensemble de l'arborescence des sous-dossiers, en mode console ;
- la quatrième ligne prend en entrée le fichier « `edt.ics` » (« `-i` » pour input) et affiche à l'écran son contenu mis en forme en mode console ;
- la cinquième ligne prend en entrée le fichier « `edt.ics` » et génère le fichier « `fragment.html` » (« `-h` » pour html ou « `-c` » pour csv) en mode console. Vous pouvez prévoir une option supplémentaire « `-p` » (« `-p` » pour page) permettant de créer un squelette de page HTML valide contenant le fragment ;
- la dernière ligne correspond au lancement de l'interface graphique.

### 3 Le jeu de données

Pour ce projet, vous devrez rechercher des exemples simples sur le web pour les fichiers de départ `.vcf` et `.ics` en vous appuyant par exemple sur les liens ci-après :

- <https://icalendar.org/>
- <https://www.w3.org/2002/12/cal/vcard-notes.html>
- <http://microformats.org/wiki/h-event>
- <http://microformats.org/wiki/h-card>
- <https://bit.ly/3F7JSYk>
- <https://support.mozilla.org/en-US/kb/how-use-virtual-card-vcard>

Pour vos tests et votre démonstration, vous utiliserez des cas d'utilisation concrets comme par exemple : une carte de visite générée à partir d'un client mail ou de votre webmail, votre edt sur l'ENT, un agenda utilisé sur votre poste ou en ligne.

N.B. : il vous appartient d'utiliser les validateurs (`vcard` / `icalendar` / `html`) pour vérifier la qualité de vos fichiers aussi bien en entrée qu'en sortie :

- <https://github.com/EBIvariation/vcf-validator>
- <https://icalendar.org/validator.html>
- <https://validator.w3.org/>

## 4 Déroulement et attendus

### 4.1 Planning

- vous identifierez les principales sous-tâches du projet à réaliser, leur niveau de priorité ainsi que le planning correspondant pour chacune d'elles (période et durée). Vous créerez un diagramme de GANTT<sup>2</sup> de votre projet. Dans tous les cas, votre diagramme de Gantt est à envoyer par mail au format **png** (capture ou export) avant le **vendredi 18 novembre** (semaine 46) : **1 point**. Ce fichier respectera les règles de nommage (voir § 4.4) ;
- nous ferons un point d'avancement le **2 décembre 2022** à distance (présence obligatoire) : **3 points** (à titre indicatif, un niveau de réalisation d'environ 50% est attendu lors de ce point d'avancement en semaine 48) ;
- vous réaliserez une vidéo sonorisée faisant office de démonstration (au plus 5 min) : **5 points** (envoi en semaine 50, **mardi 13 décembre** 23h30 au plus tard). Cette démonstration permettra de visualiser les deux fonctionnements en mode console et graphique (il est utile de prévoir un scénario...). Ce fichier respectera les règles de nommage (voir § 4.4) ;
- tous les autres attendus (voir § 4.3) sont à déposer en semaine 50 le **lundi 12 décembre** 23h30 au plus tard ;
- une soutenance de 10 min par personne (5 minutes de présentation et 5 minutes de question) est prévue le **vendredi 16 décembre**, à partir de 11h30 et jusqu'à 16h00 au plus tard. Un ordre de passage avec la salle vous sera communiqué plus tard.

### 4.2 Soutenance (4 points)

- la diapositive de titre présentera le contexte, le sujet. (i.e. : la page de garde sera compacte) ;
- les autres diapositives (5 au plus) devront présenter les spécificités de réalisation du projet et les principaux éléments de conception, donc aucune information "évidente" (ex. détail du sujet, progression personnelle, ...) ne devra être mentionnée ;
- la diapositive de conclusion mettra en évidence le niveau d'achèvement du projet (points traités et non traités du cahier des charges et extensions s'il y en a) ;
- vous devrez prévoir une version PDF de votre diaporama au cas où.

NB : à éviter ABSOLUMENT : les diagrammes de classes UML illisibles (trop chargés, ...), les programmes (code Python), les captures d'écran (puisque'il y a aussi une vidéo de démonstration), la liste des outils utilisés, ...

Important : vous devrez avoir votre machine portable allumée, prête avec l'ensemble des logiciels nécessaires à votre présentation déjà lancés AVANT d'entrer dans la salle. Votre portable devra disposer d'un port VGA, HDMI ou USB-C ou vous devrez prévoir un adaptateur correspondant à votre situation. Vous aurez aussi sur vous une clé USB contenant la version PDF de votre présentation...

### 4.3 Attendus (7 points)

- complétude et qualité du projet : **3 points** (code Python commenté ET documenté, gestion des librairies fonctionnelles, arborescence des différents documents). Les livrables sont à déposer sur la plate-forme pédagogique au plus tard le **lundi 12 décembre** à 23h30 **ET à partager via un lien wetransfer...**
- fichier « README.txt » contenant les informations de base ainsi que les informations spécifiques utiles (**1 point**) ;

---

2. <https://www.ganttproject.biz/download/free>

- rapport de projet (minimum 5 pages, maximum 10 pages) : **3 points** (le fond et la forme seront évalués). Les 2 fichiers suivants sont à rendre :
  1. le document de traitement de texte (format odt) ;
  2. la version pdf de votre rapport ;
- l'ensemble des fichiers sources (.py) du projet (sans aucune référence à votre outil de développement) ;
- et la documentation (si possible au format Doxygen, à défaut au format Pydoc).

#### 4.4 Règles de nommage

Les fichiers doivent respecter le nommage suivant `L3_I-NOM_Prenom-xx.zip`, où `xx`, par exemple, sera remplacé par **Gantt** en réponse à la question 1 du §4.1 !

Lors du rendu final, **tous les fichiers et sous-dossiers** à remettre doivent être placés dans **un répertoire unique portant vos nom et prénom** (sous la forme `NOM_Prenom`) à compresser en un seul fichier au **format zip** qui sera déposé sur la plate-forme pédagogique de cours (moodle) et doivent respecter la hiérarchie ci-dessous :

```
NOM_Prenom
|__ doc_
|    |_ demonstration
|    |_ diaporama
|    |_ rapport
|
|__ src_
|    |_ cli
|    |_ gui
|
|_ README.txt
```

Le non respect de ces règles entraînera des points de pénalité !