

On souhaite réaliser un gestionnaire de *contacts* (un carnet d'adresse).

1. Contact

Un contact est un organisme ou une personne (qui peut elle même faire partie d'un ou plusieurs organismes). Une personne est définie par un nom, éventuellement suivi par d'autres champs qui caractérisent cette personne (sexe, tel, adresse, email, ... ou tout autre champ que vous jugerez pertinent). Un organisme est défini par son nom, sa raison sociale, ses coordonnées, éventuellement suivis par d'autres champs que vous jugerez pertinents. Certains champs sont constitués d'un seul élément, alors que d'autres sont constitués de plusieurs éléments (éventuellement de types différents). Voir l'annexe page 2 pour une liste (non exhaustive) des champs associés à une personne et un organisme.

2. Gestionnaire de contacts

Le programme doit pouvoir gérer plusieurs contacts, il doit donc pouvoir :

- Afficher une liste triée des contacts (en sélectionnant les critères de tri).
- Afficher un contact sélectionné dans cette liste.
- Editer un contact sélectionné dans cette liste.
- Créer un nouveau contact.
- Détruire un contact.
- Rechercher des contacts satisfaisant un ou plusieurs critères.
- Enregistrer la liste des contacts dans un fichier :
 - Dans un format défini par vous même (en XML par exemple)
 - Dans un format plus standard comme vCard (2.1, 3.0 ou 4.0) ou xCard.
- Lire (ou ajouter) une liste de contacts depuis un fichier :
 - Depuis un format défini par vous même (voir ci-dessus)
 - Depuis un format plus standard (vCard et/ou xCard)

3. Travail demandé

Ce projet est à faire en binôme en utilisant l'API QT pour l'interface graphique.

Implémentez un programme de gestion de contacts en respectant l'architecture MVC (Modèle / Vue(s) / Contrôleur(s)). Vous veillerez donc à séparer clairement le modèle (la liste des contacts) des vues (l'affichage de la liste et des contacts) et des contrôleurs (les widgets permettant de modifier les vues et/ou le modèle). Chaque classe devra être documentée (avec Doxygen) et son auteur identifié.

- 1) Concevez les classes permettant de représenter les contacts et leurs champs associés ainsi que la collection de contacts. Les classes directement en relation avec l'interface graphique pourront être implémentée sous la forme de QObjects afin de fournir des slots à l'interface et des signaux vers l'interface.
- 2) Concevez l'interface graphique en QT permettant de gérer vos contacts.
- 3) Concevez les classes nécessaires à l'importation et l'exportation de vos contacts dans divers formats :
 - a) Votre propre format.
 - b) Les formats vCard.

Barème (à titre indicatif)

- Hiérarchie de classes représentant le modèle (5 points)

- Fonctionnalités implémentées (10 points)
- Interface graphique (5 points)

Annexes

Champs des contacts :

Liste (non exhaustive) des champs associés à une personne :

Champ	Type	arité	Sous-champ	Type	arité	Exemple
nom	structuré	1	Nom	texte	1	Roussel
			Prénom	texte	1..*	David
			Surnom	texte	0..1	
			Préfixe	texte	0..*	Dr.
			formatage	structure	1	Dr. David Roussel
sexe	enum : {homme, femme, ...}	0..1				homme
tel	structuré	0..*	type	enum : {work, home, ...}	1	work
			type tel	enum : {fixe, cellulaire, ...}	1	fixe
			type données	enum : {voice, data, ...}	1	voice
			n°	texte ou structure	1	01 69 36 74 62
adresse	structuré	0..*	type	enum : {work, home, ...}	1	work
			rue	texte	1..*	1 place de la Résistance
			zipcode	texte ou structure	1	91025
			localité	texte	1	Evry cedex
			Région	texte	0..1	
			pays	texte	1	France
			geo	latitude + longitude	0..1	48.626512, 2.432210
email	structuré	0..*	nom	texte	1	david.roussel
			domain	texte	1	ensiie
			extension	texte	1	fr
site	structuré	0..*	type	enum : {web, facebook, go	1	web
			url	url	1	http://www.ensiie.fr
photo	image	0..1	type	enum : {JPEG, GIF, ...}	1	
			url	url	1	
organisation	card	0..*				card de l'ENSIIE
type	texte ou enum	0..*				collègue, ami
date MAJ	timestamp	1				
note	texte	0..*				

Liste (non exhaustive) des champs associés à un organisme :

Champ	Type	arité	Sous-champ	Type	arité
nom	structuré	1	Nom	texte	1
			raison sociale	...	0..1
		
tel	structuré	0..*	type tel	enum : {fixe, cellulaire, ...}	1
			type données	enum : {voice, data, ...}	1
			n°	texte ou structure	1
adresse	structuré	0..*	intitulé	texte	0..1
			rue	texte	1..*
			zipcode	texte ou structure	1
			localité	texte	1
			Région	texte	0..1
			pays	texte	1
			geo	latitude + longitude	0..1
email	structuré	0..*	nom	texte	1
			domain	texte	1
			extension	texte	1
site	structuré	0..*	type	enum : {web, facebook, google+, ...}	1
			url	url	1
logo	image	0..1	type	enum : {JPEG, GIF, ...}	1
			url	url	1
membre	structuré	0..*	card	card d'une personne	1
			fonction	texte	1
date MAJ	timestamp	1			
note	texte	0..*			

Documentation

- Documentation QT : <http://qt-project.org/doc/qt-4.7/>
- Documentation vCard : <http://en.wikipedia.org/wiki/VCard>