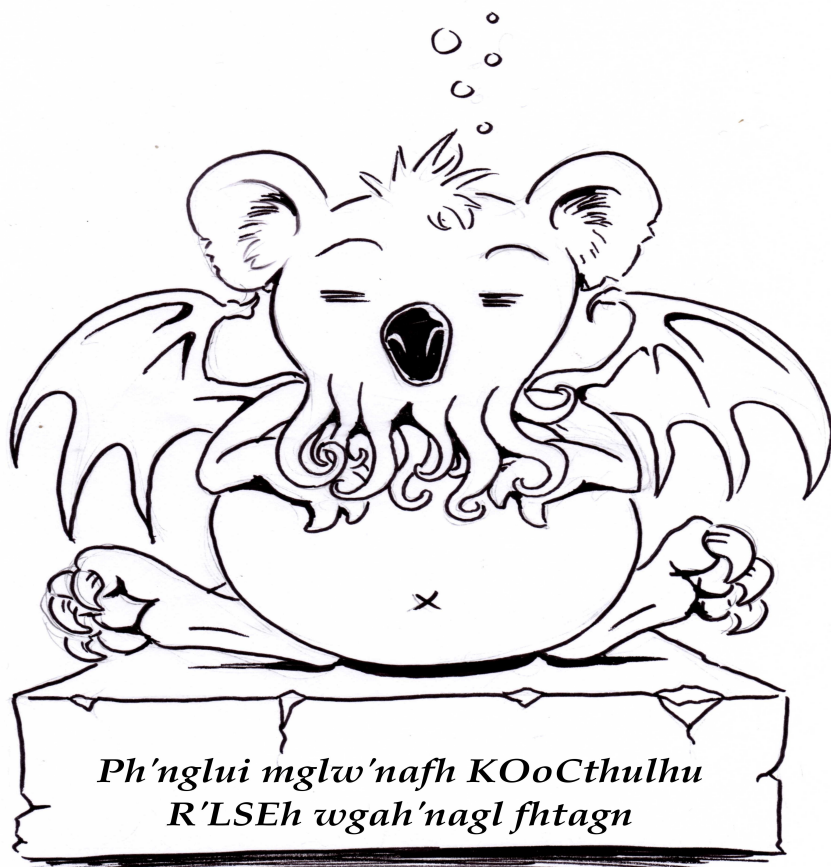


KOOC - SUJET



CALL OF KOOCTHULHU

*«The most merciful thing in the world, I think, is the inability of the human mind to correlate all its contents. We live on a placid island of ignorance in the midst of black seas of infinity, and it was not meant that we should voyage far. The sciences, each straining in its own direction, have hitherto harmed us little; but some day the piecing together of dissociated knowledge will open up such terrifying vistas of reality, and of our frightful position therein, that we shall either go mad from the revelation or flee from the deadly light into the peace and safety of a new dark age.» **

Le temps des dévotions et des sacrifices au grand ancien des koalas est venu.

Oserez-vous relever le défi ?

*une bonne connaissance de l'oeuvre de H.P.Lovecraft est nécessaire pour comprendre les références poétiques de ce sujet.

RENDUS :

Lot programme

Répertoire de rendu : `~/rendu/kooc`

Vous devez fournir un programme nommé 'kooc' (à la racine du répertoire de rendu) et ses bibliothèques (hors pyrser et cnorm) écrit en Python 3.3+ capable de pré-processer le langage KOOC comme défini dans l'ensemble des cours (disponible sur l'e-learning).

Les 3 premières parties clairement spécifiées dans les cours (partie 1 à 3) représentent une partie obligatoire. L'implémentation de la partie 4 et 5 représente du bonus.

VOUS DEVEZ UTILISER LES TECHNIQUES D'HERITAGE DE GRAMMAIRE DE PYRSER

c'est-à-dire :

- **UTILISEZ LE CNORM**
- **IL EST INTERDIT DE MODIFIER CNORM**

Lot documentation

Vous devez fournir une documentation de **CONCEPTION** de l'ensemble du projet. La conception porte donc sur l'intégralité de la partie obligatoire (1,2,3) + vos bonus. La conception ne représente pas ce que vous AVEZ codé mais ce que vous DEVEZ coder. C'est les plans de votre maison. On ne trace pas le plan d'une maison en constatant ce qui a été bâti (« Oui monsieur, les murs ne sont pas droits comme indiqué dans le plan. ». Désolé, j'achète pas la maison, moi).

La documentation de conception n'est pas facultative. Sans conception vous ne pourrez pas réaliser ce projet. Sans conception pas de note !

Schémas, diagrammes et algorithmes sont les bienvenus. L'aspect 'administratif' de la documentation (cartouche, suivi qualité etc.) est un minimum mais n'est clairement pas suffisant. C'est le contenu qui sera regardé et la manière dont vous expliquez les choses. Chaque membre du groupe doit comprendre grâce à la documentation ce qu'il doit coder.

Concevez cette documentation comme un document de travail que vous vous échangez entre vous. Quelqu'un n'ayant pas travaillé sur le sujet doit aussi comprendre ce que vous faites. Toutefois ne reformulez pas le cours ou le sujet.

Citez-le en référence.

Vous vous adressez à un ingénieur (travaillant sur le projet, ou qui continuera le projet) pas à un commercial, ni à l'utilisateur final! Les commerciaux ne lisent pas

les documents de **conception** :P. Vos réflexions sur brouillon mises au propre peuvent y figurer en précisant bien qu'il s'agit de notes de conception. La critique de vos choix techniques est un plus appréciable dans ces notes. Vous pouvez essayer de présenter plusieurs pistes de réflexions, et ensuite expliquer vos choix. En cas d'échec cela permet de comprendre pourquoi. Pourquoi avons-nous choisi telle méthode plutôt que telle autre ? Etc.

Expliquez dans le détail ce que vous avez compris du sujet (étude des besoins) et présentez votre solution.

Considérations administratives

de 3 à 6 étudiants

Rendu dans le répertoire ~/rendu/**kooc** du chef de groupe.

Vous devez faire valider votre documentation auprès du comité de lecture aux adresses suivantes (1 mail, 2 adresses en CC afin d'avoir un seul thread):

- raphael.geneteix@gmail.com (Raphael Geneteix) pour l'expression écrite
- lionel@lse.epita.fr (Lionel Auroux) pour le contenu
- Le mail du DPR de votre région (ou responsable du module KOOC dans votre région).

Envoyez votre doc de conception KOOC au format PDF.

Le sujet du mail DOIT ETRE "[KOOC_techX_20XX] loginduchefdegroupe_x" sinon il ne sera pas traité sans vous prévenir.

Le fichier DOIT ETRE «[KOOC_techX_20XX] loginduchefdegroupe_x.pdf».

Afin de vous aider à réaliser votre documentation de conception les 3 premiers suivis du projet KOOC sont consacrés à la compréhension du cours, et à la rédaction de votre documentation. Vous devrez montrer une documentation complète au **3ième suivi**.

TOUS LES SUIVIS DE KOOC SONT OBLIGATOIRES.

La note de la note de conception compte pour + de 40 % de la note finale. Elle est constituée à 50 % d'une note d'expression écrite et à 50 % d'une note d'algorithmie (fiabilité technique des algos/schémas présentés).

Le premier draft de la documentation doit être envoyé **le soir du 3ième suivi à 23h42**. Vous acquerrez 80-100 % de la note d'expression écrite sur cet envoi.

Le comité de lecture vous soumettra alors un premier compte rendu de relecture avec quelques modifications éventuelles.

Un aller-retour s'effectuera donc entre votre groupe et le comité de lecture jusqu'à la date de soutenance.

Une liste des documentations validées par le comité de lecture sera communiquée à l'équipe KOOC afin de valider les groupes pouvant passer leur soutenance FINALE.

SEULS LES GROUPES FAISANT PARTIE DE CETTE LISTE SERONT AUTORISÉS À PASSER LEUR SOUTENANCE.

Chaîne de production

fichier source KOOC *.kc

fichier header KOOC *.kh

- nom de la librairie runtime (si nécessaire):

- -statique "libkrt.a"
- -dynamique "libkrt.so"

- nom du fichier header de macro et/ou fonction réutilisable (si nécessaire) que vous pouvez choisir d'inclure automatiquement: kooc.h

- Vous pouvez avoir besoin d'utiliser des fichiers intermédiaires. Dans ce cas, ils doivent avoir une extension commençant par 'k'.

- Vous devez utiliser **cpp** pour la phase de 'pré-processing' avant kooc. **cpp** est le nom de l'exécutable du C pré-processeur. Vous pouvez aussi ré-implémenter **cpp** en python/pyrser(bonus).

- Exemples d'utilisation (tous ces cas doivent marcher):

```
# ls
test.kc test.kh
# kooc test.kh
# ls
test.h test.kc test.kh
# kooc test.kc
# ls
test.c test.h test.kc test.kh
# cd somewhere/in/the/disk
# ls
myProgramme.kc myLib.kh myLib.kc
# /la/ou/ce/trouve/kooc/kooc myProgramme.kc
# ls
myProgramme.c myProgramme.kc myLib.kh myLib.kc
#
```

- Compilations :

```
# ls
mod1.kc mod1.kh mod2.kc mod2.kh mod3.kc mod3.kh
# for file in `ls`; do kooc $file; done
```

```
# gcc -I${la_ou_vous_avez_mis_votre_kooc.h} *.c \  
-L${la_ou_vous_avez_mis_la_lib} -lkrt -o test
```

Le Sujet

Il s'agit d'implémenter kooc en tenant compte des exigences suivantes :

- Code généré
Le code généré peut utiliser certaines fonctionnalités spécifiques à GCC (C Extension, __attribute...) Par exemple pour l'implémentation des mécanismes d'exceptions (mot-clefs : @try/@catch/@throw).
<http://gcc.gnu.org/onlinedocs/gcc-X.X.X/gcc/>
- Contraintes de conception
Vous pouvez rajouter de nouvelles fonctionnalités. Ces fonctionnalités seront documentées dans la doc de conception. Il est obligatoire pour ces fonctionnalités d'utiliser de nouveaux mot-clefs ou du sucre syntaxique qui ne perturberont pas le bon fonctionnement de la partie obligatoire. Toutes ces nouvelles fonctionnalités sont du pur bonus au niveau barème. Voici une liste non-exhaustive de quelques fonctionnalités bonus:
 - Génération de makefile
cf gnatmake dans le langage ADA.
 - CPP in python/pyrser
 - Héritage avancé type C++
Héritage multiple de la partie 5 du cours.
 - Héritage avancé type Ruby
Interface abstraite, Module mixin de la partie 5 du cours.
 - Notion de private/public/protected sur membre des classes. Possibilité d'étendre le concept avec héritage multiple/module mixing
 - Accesseurs
Générer des pseudo-fonctions Get et/ou Set pour accéder aux variables membres directement. google C#
 - Dynamic binding
Résolution tardive d'un appel de méthode via sélecteurs. google objective C
 - Dynamic classing
Modification dynamique d'une classe, ajout de méthode dynamiquement etc... google dynamic classing et cf professeur
 - Generics ou Template
Generics comme en Java ou Template comme en C++. Cependant, vous pouvez n'en avoir qu'une version réduite, ou alors un concept nouveau. cf professeur

- Librairie standard

Listes simplement chaînées, doublement chaînées, arbre binaire, table de hash. Tout cela codé avec votre KOOC! :)

Vous pouvez discuter de ces fonctionnalités avec votre professeur.

Notamment pour évaluer la pertinence et la complexité de la fonctionnalité.

Vous serez noté en priorité sur l'implémentation correcte de KOOC. Les fonctionnalités en plus sont du bonus. Vous ne serez donc pas pénalisé si vous n'implémentez pas d'autres fonctionnalités (note jusqu'à 20). Un KOOC bancal avec quelques fonctions bonus, pourquoi pas. Mais vous serez limité dans les choix si votre KOOC est trop faillible. Et finalement, un KOOC complet (fonctions des TPs OK) avec des fonctionnalités en plus dépassera 20.