Conception à base de patrons II

1- Objectifs

Ce laboratoire permet aux étudiants de se familiariser avec l'implémentation des patrons de conception "Template Method" et "Factory Method". Cette implémentation est effectuée à l'aide de visual studio code en C++. Un cadriciel vous est fourni, ainsi que des tests déjà implémentés permettant de vérifier le bon fonctionnement de votre code. Une fois le code complété, vous pourrez ainsi vous assurer que son comportement est correct. On vous suggère d'ailleurs d'aller lire les fichiers de tests afin de comprendre ce qui est demandé, et cela peut vous aider à répondre aux questions.

De plus, cette fois on vous recommande fortement de compléter les patrons dans l'ordre, soit de commencer par Template Method, puis terminer avec Factory Method, qui dépend de l'implémentation de Template Method.

Ressources supplémentaires vers les patrons de conception

- Template Method:

Design Patterns: Template Method in C++

- Factory Method:

Design Patterns: Factory Method in C++

2- Patron Template Method (20 points)

Des objets représentant des arrosoirs ont été ajoutés au logiciel Polyjet. Ceux-ci possèdent tous la méthode `activer(int debit, int duree)` qui partage toujours la même logique, c'est à dire retourner une quantité d'eau qui correspond à `arroser(debit) * duree`. Par contre, la fonction `arroser(int debit)` change selon la sorte d'arrosoir. Cette méthode est déjà implémentée pour l'arrosoirFixe dans ArrosoirFixe.cpp en guise d'exemple. Les classes qui participent à ce patron sont:

- `AbsArrosoir`
- `ArrosoirFixe`
- `ArrosoirOscillant`
- `ArrosoirRotatif`

Remplissez les méthodes dans les fichiers AbsArrosoir.h, AbsArrosoir.cpp, ArrosoirOscillant.cpp et ArrosoirRotatif.cpp.

3- Patron Factory Method (40 points)

Dans le cadre de ce travail, on considère que les parcelles peuvent être configurées selon une liste d'arrosoirs. Une "configuration" correspond donc à une liste d'arrosoirs. On aimerait pouvoir facilement utiliser n'importe quelle configuration sur une parcelle. Pour ce faire, nous utiliserons le patron Factory Method, qui permettra de créer des objets AbsArrosoir, les mettre dans une liste, puis d'obtenir la quantité d'eau générée par ces arrosoirs. Les classes qui participent à ce patron sont:

- `AbsArrosoir` (et toutes ses classes enfants)
- `Parcelle`
- `AbsConfigurationArrosage`
- `ConfigurationArrosageA`
- `ConfigurationArrosageB`
- `ConfigurationArrosageC`

Remplissez les méthodes dans les fichiers AbsConfigurationArrosage.cpp, ConfigurationArrosageB.cpp, ConfigurationArrosageC.cpp et Parcelle.cpp .

4- Questions (40 points)

Question 1 (10 points)
Faites 2 diagrammes de classes du logiciel, un pour chaque patron. Indiquez les attributs, les méthodes et les relations entre les classes. Vous pouvez utiliser un autre logiciel qu'Enterprise Architect pour les faire. Vous devez inclure vos 2 diagrammes dans votre fichier tp5.pdf. (voir la section 5- Remise)

Question 2 (5 points)
Que faudrait-il modifier pour avoir un comportement "par défaut" pour les arrosoirs, afin que l'on puisse en créer de nouveaux types sans avoir à préciser une logique pour la fonction `arroser()`?

Question 3 (10 points)
Dans le patron factory method, qu'est-ce qui joue le rôle de "code client", de "créateurs concrets" et de "produits concrets"? Quelle méthode joue le fameux rôle de "factory method"?

Question 4 (10 points)
Comment le patron factory method permet au code client de pouvoir ignorer le type exact des produits concrets?

Question 5 (5 points)
Expliquez comment le patron factory method assure une bonne gestion des ressources dans le code que vous avez complété.

5- Remise

Créez un fichier compressé (.zip) nommé LOG2410_MatriculeA_MatriculeB_TP5.zip. Celui-ci devrait comprendre:

- Un dossier nommé "src" qui contient tous les fichiers .cpp modifiés
- Un fichier nommé "tp5.pdf" qui contient les réponses aux questions. Celui-ci doit inclure les diagrammes de classes de la question 1.

Remettez le fichier compressé sur Moodle avant la date et l'heure de remise.

Annexe - Informations supplémentaires

(Informations provenant des posts d'Ahmed dans le discord)

Configuration sous Windows: VSCode

Pour ceux qui aiment avoir une distribution Linux installée avec leur Windows : Installation wsl2 https://docs.microsoft.com/en-us/windows/wsl/install-win10, c'est un hyperviseur qui vous offre une intégration transparente entre Windows et Linux, des temps de démarrage rapides, une faible empreinte de ressources et ne nécessite aucune configuration ou gestion de machine virtuelle.

Pour ceux qui veulent compiler et debugger sous Windows: Configuration winGW pour compiler et debugger votre code sous windows
https://code.visualstudio.com/docs/cpp/config-mingw

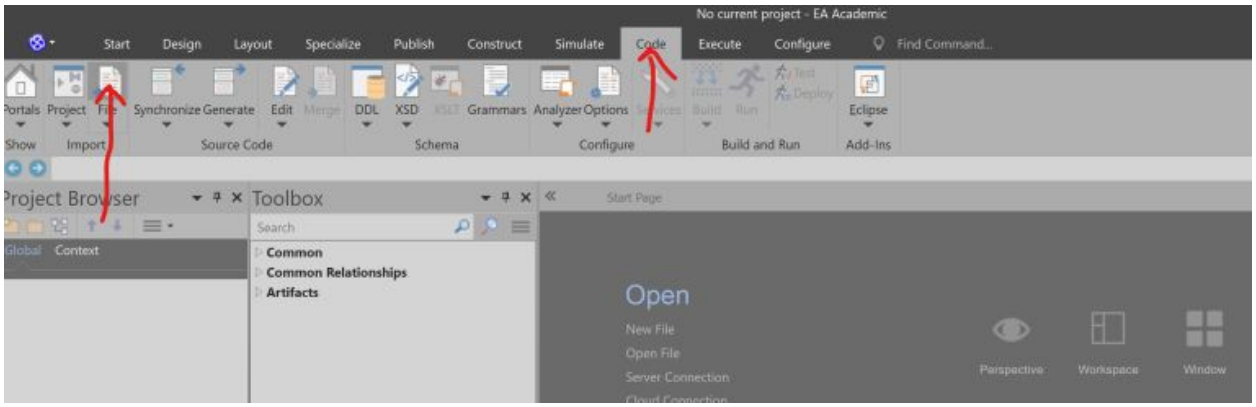
Utilisez les même fichiers fournis avec votre TP figurant sous .vscode juste changer les paths de votre compiler et debugger à ceux fraîchement importés par minGW (normalement dans le launch.json vous devez changer
"miDebuggerPath": "/usr/bin/gdb"
par
"miDebuggerPath": "C:\Program Files\mingw-w64\x86_64-8.1.0-posix-seh-rt_v6-rev0\mingw64\bin\gdb.exe"
et dans le tasks.json le
"command": "/usr/bin/g++",
par
"command": "C:\Program Files\mingw-w64\x86_64-8.1.0-posix-seh-rt_v6-rev0\mingw64\bin\g++.exe",
).

N'oubliez pas de supprimer l'exécutable polyjet ou bien de changer la commande du fichier launch.json de
"program": "${workspaceFolder}/polyjet",
à
"program": "${workspaceFolder}/polyjet.exe",

Pour ceux qui veulent avoir toutes les commandes Linux sur leur Windows:
https://cmder.net/

Enterprise Architect

Vous pouvez utiliser Enterprise Architect pour comprendre les relations entre vos différentes classes. Entreprise Architecte peut *analyser le code C++* et vous *générer automatiquement vos diagrammes de classes avec des relations entre celles-ci*. Pour ce faire, allez dans le menu code, et choisissez file et importez les fichiers .h. Vous allez trouvez qu'il a parsé toutes les classes des .h. Créer un nouveau diagramme de classe et en drag and drop toutes les classes, vous allez trouver qu'au fur et à mesure que vous ajoutez des classes les liens entre celles-ci s'ajoutent avec les autres. **N'oubliez pas que cela reste un outil et que vous devriez faire attention aux relations qui peuvent exister entre les classes** (aggrégation , composition ...) **dans vos réponses aux questions concernant les diagrammes de classe.**



Extensions VSCode

- ms-vscode

Extensions supplémentaires intéressantes: