**M2 IF Apprentissage projet composants**

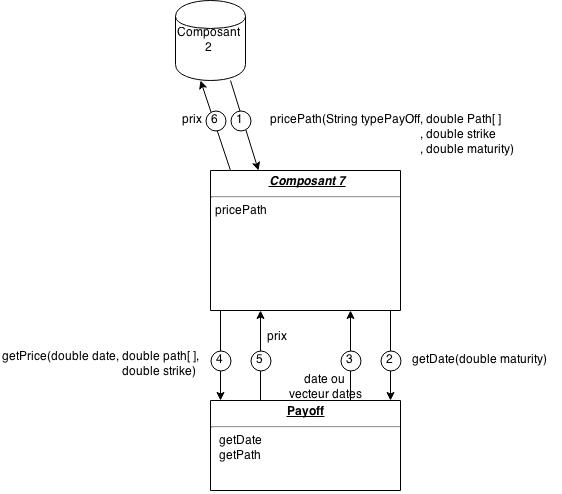
Spécifications composant 7

Echantillonnage Invocation du payoff Delta KT

|  |
| --- |
| **Groupe 6** |
| Hamza ZABIN |
| Aymen AMAMOU |
| Juanshu HAN |
|  |

|  |  |  |  |
| --- | --- | --- | --- |
| Version doc | Date | Auteur(s) | Modifications |
| 1.0 | 23/02/2015 | Zabin & Han & Amamou | Version initiale |
| 2.0 | 23/02/2015 | Zabin & Han & Amamou | Specification génerale |
| 2.1 | 24/02/2015 | Zabin & Han & Amamou | Finalisation de la spécification |
| 3.0 | 07/04/2015 | Zabin & Han & Amamou | Modification |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

1. **Description générale**
   1. **Architecture du module**



* 1. **Le contrat**

Un contrat contient une ou plusieurs clauses dépendant du type du contrat.

Dans un premier temps, on modélisera un contrat de type call. Ce contrat est d’une période de 2ans, donc contient 504 valeurs de spot et une date de maturité

* 1. **L’échantillonneur**

L’échantillonneur interroge le contrat et obtient la date (dans le cas d’un call) ou le vecteur dates du contrat. Il obtient aussi à partir du composant 6 le chemin aléatoire. Et échantillonne le chemin en fonction des dates.

Ensuite, l’échantillonneur appelle le contrat une seconde fois avec la date et le spot correspondant, et renvoie le prix du contrat.

Ce prix va être utilisé par la suite dans le composant 2.

1. **Description des fonctions**

Dans un premier temps, on s’intéressera au contrat call. Celui-ci sera modélisé en une classe du nom PayOff

* **La méthode getDate :**

Cette méthode va renvoyer une date ou un vecteur dates dépendant du type du contrat. Si celui-ci est un call, elle renverra directement la maturité.

* **La méthode getPrice :**

Cette méthode va prendre en paramètre:

* Le strike ***K*** de type double***.***
* La date de type double déjà calculée par la méthode getDate
* Le path de type double[].

Cette fonction va calculer le prix du Strike à l’aide de la formule Prix = Max(S-K, 0). Où S est le spot calculé à partir du path.

Elle va renvoyer le prix ***P*** de type double.

* **La fonction de l’échantillonneur (pricePath):**

Cette fonction prend en paramètre:

* Le chemin aléatoire ***path***.
* Le strike ***K***.
* La maturité ***maturity.***
* Le type du contrat ***typePayOff.***( une chaine de caractère, « call » dans notre exemple)

Cette fonction va dans un premier temps stocker la (ou les) date(s) selon le type de contrat, et ceci en appelant *getDate* implémenté dans la classe contrat.

Pour chaque date dans le vecteur des dates, on stockera dans un autre vecteur le spot correspondant dans le chemin aléatoire. (Une seule date et un seul spot dans le cas du Call)

Ensuite, on appelle *getPrice* sur le(s) spot(s) calculé(s).

La fonction renvoie:

* Le prix ***P*** du contrat échantillonné.

Veuillez-vous assurer à ce que la fonction échantillonneur soit générique, c’est à dire acceptant n’importe quel type de contrat. Dans un premier temps nous nous focalisons sur le contrat Call. On aura donc une seule date et donc un seul spot. Mais la fonction devrait aussi traiter un vecteur de dates (qui sera extrait d’un autre contrat) et produire un vecteur de spot.

1. **Les prototypes :**

En plus du contrat, on dispose de 3 fonctions à implémenter, comme vu au-dessus. Voici donc les prototypes de ces fonctions :

Double pricePath( String typePayOff, double path[], double strike, double maturity);

Double getDate(double maturity);

Double getPrice(double date, double strike, double path[]);