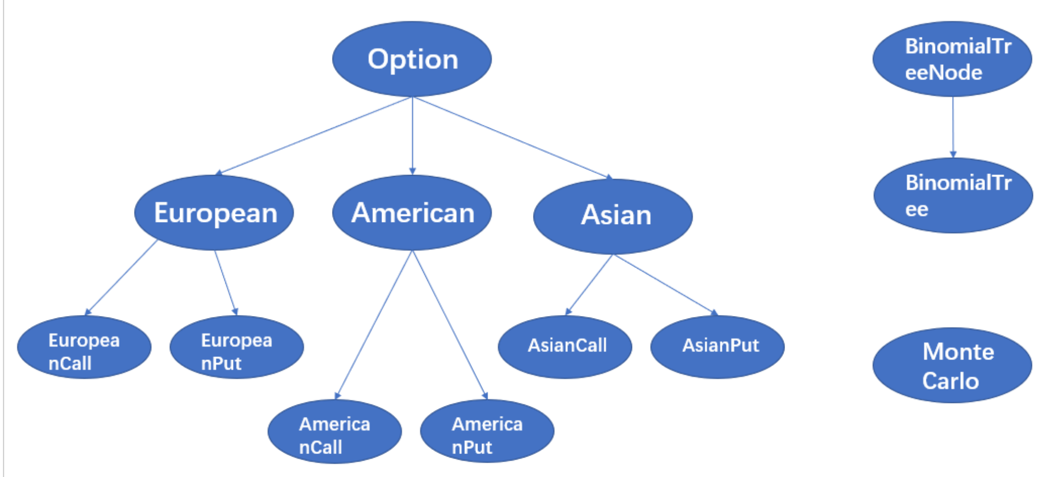
代码说明

这段代码的目的是用二叉树，偏微分方程（PDE），蒙特卡洛模拟，这三种方法分别给欧式、美式、亚式期权定价。本程序中各个类的关系如下图

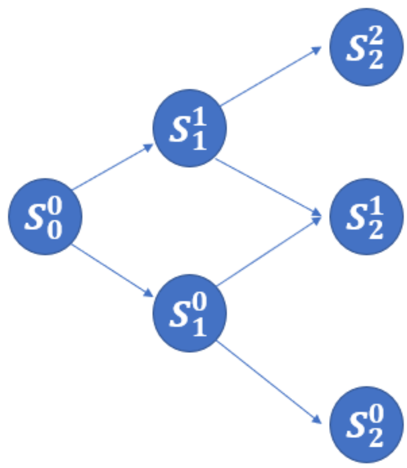


“Option”类用于初始化基本数据，如利率，波动率，到期日，股价等；“European”“American”和“Asian”是最核心的部分，用不同的方法进行定价；图中最下层的类用于存储不同的 payoff 函数。右边的类是用于构建二叉树的，对于欧式和美式期权，只需要调用“BinomialTree”类，这个类中会产生二叉树算法所需要的 u，d，p 参数。而亚式期权的二叉树算法需要追踪路径，所以专门用“BinomialTreeNode”类来记录每个点，点与点之间用指针相连，后面会详细讲解。

1、二叉树算法

（1）欧式期权

在类“European”中，先调用类“BinomialTree”的函数 GetSharePrice，获得最尾端的股票价格；然后用 EuropeanPayoff 函数获得对应的期权价值，储存在向量 currentPrice 中。之后的 for 循环就是从二叉树的尾端向头部迭代，所以向量 currentPrice 不断更新，直到得到 t=0 时的期权价格。



（2）美式期权

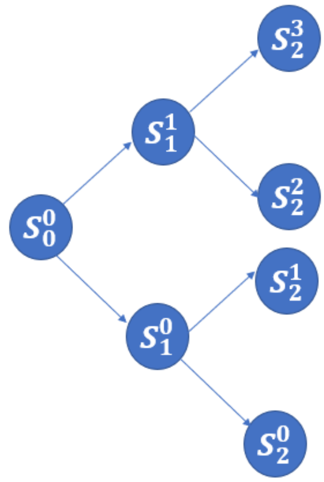
美式期权的做法和欧式期权非常相似，唯一的区别是在每个节点处，都要判断是否立即行权，这体现在 AmericanPayoff 上，并且在更新向量 currentPrice 时，都要对是否立即行权进行判断。

（3）亚式期权

因为与路径有关，所以我们使用指针来追踪股价生成的路径。

先构造了一个结点类“BinomialTreeNode”，用于生成每一个结点，包含两个元素，结点的数值和指向上一个结点的指针 parent。类“BinomialTree”继承于“BinomialTreeNode”，其中函数“MakeTreeNode”用于生成一个新的结点。

在类“Asian”的函数PriceAsianOptionWithBinomialTree()中，分为三步求解，都是以层为单位，处理完一层的所有结点再处理下一层。第一步，生成二叉树，各个结点间有指针联系；第二步，求出二叉树最底层每个结点对应的路径均值；第三步，类似欧式期权，求出 t=0 时对应的期权价值。具体细节见代码注释，我写的很详细。



二、偏微分方程（PDE）算法

（1）欧式期权

欧式期权的 PDE 是

$$\frac{\partial V}{\partial t}+rS\frac{\partial V}{\partial S}+\frac{1}{2}\frac{\partial^2 V}{\partial S^2}\sigma^2 S^2=rV$$

其中 V 是期权价值，S 是标的物价格，r 是利率，σ^2是波动率。当 S 的取值范围为[S1,S2]时，对于看涨期权边界条件是

$$V(S,T)=\max(S-K,0)$$

$$V(S1,t)=V(S2,t)=K$$

对于看跌期权边界条件是

$$V(S,T)=\max(K-S,0)$$

$$V(S1,t)=V(S2,t)=0$$

下面我们对这个 PDE 做一些变换

$$\frac{\partial V}{\partial t} = rV - rS\frac{\partial V}{\partial S} - \frac{1}{2}\frac{\partial^2 V}{\partial S^2}\sigma^2 S^2$$

$$\Leftrightarrow \frac{\partial V}{\partial t} = rV - rS\frac{V(S+\Delta S) - V(S)}{\Delta S} - \frac{1}{2}\sigma^2 S^2 \frac{V(S+\Delta S) + V(S-\Delta S) - 2V(S)}{\Delta S^2}$$

如果我们定义$S = \Delta S * n$, $V_n(t) = V(\Delta S * n, t)$，则上式等价于

$$\frac{V(t) - V(t-\Delta t)}{\Delta t} = rV_n - rn(V_{n+1} - V_n) - \frac{1}{2}\sigma^2 n^2 (V_{n+1} + V_{n-1} - 2V_n)$$

即

$$V(t-\Delta t) = V(t) - \Delta t * (rV_n - rn(V_{n+1} - V_n) - \frac{1}{2}\sigma^2 n^2 (V_{n+1} + V_{n-1} - 2V_n))$$

接下来的编程过程见代码注释。

（2）美式期权

美式期权的偏微分算法与欧式期权完全一致，唯一的区别是每得到一个新的期权价格时都要判断一下是否立即行权。

三、蒙特卡洛算法

（1）欧式期权

蒙特卡洛的基本思想是：

- 对 S 的随机路径进行抽样
- 计算期权的收益，
- 重复前两步从而获得许多收益样本，计算这些样本的均值
- 对该均值进行贴现，结果即为期权价格

计算 S 值的公式为

$$S(T) = S(0)\exp\left[\left(r - \frac{\sigma^2}{2}\right)T + \sigma\varepsilon\sqrt{T}\right]$$

其中$\varepsilon \sim N(0,1)$，由蒙特卡洛模拟产生。

（2）美式期权（最小二乘蒙特卡洛法）

因为每天都需要判断是否应该立即行权，所以我们采用“最小二乘蒙特卡洛法”，原理比较复杂。对于看跌期权，本方法仅适用于 shareprice>strike 的情况，而对于看跌期权，本方法仅适用于 shareprice<strike 的情况。

该方法分为三步，第一步是进行 N 次蒙特卡洛模拟，得到一条 N+1 个点的股票价格样本路径，重复 M 次，则得到 M 条股票价格样本路径。举个例子，模拟美式看跌期权，取 M=10，N=3，随机生成的路径如下

	t=0	t=1	t=2	t=3
1	1.00	0.98	0.93	0.95
2	1.00	1.09	1.18	1.11
3	1.00	0.96	1.02	0.98
4	1.00	0.98	0.97	0.94
5	1.00	0.97	0.9	0.88
6	1.00	0.94	0.93	0.91
7	1.00	0.99	0.99	0.99
8	1.00	0.99	1.01	0.86
9	1.00	1.08	1.07	1.05
10	1.00	0.99	1.02	1.08

取执行价 K=1.05，那么在 t=3 处各条路径的价值为 max(K-S,0)，折现到 t=2 处，记做 Y；把“t=2 时的股票价格”视为 X

	Y	X
1	0.10×0.99584	0.93
2		
3	0.07×0.99584	1.02
4	0.11×0.99584	0.97
5	0.17×0.99584	0.90
6	0.14×0.99584	0.93
7	0.06×0.99584	0.99
8	0.19×0.99584	1.01
9		
10	0.00×0.99584	1.02

进行最小二乘法拟合得到

$$E\left[Y \mid X\right] = -0.093 + 1.083X - 0.908X^2$$

将 X 代入，得到下表，第二列是立即行使期权的收入，第三列是继续持有期权的收入，反复迭代直到 t=0。

1	0.12	0.13
2		
3	0.03	0.07
4	0.08	0.10
5	0.15	0.15
6	0.12	0.13
7	0.06	0.09
8	0.04	0.08
9		
10	0.03	0.07

最后我们可以得到每条路径的最优行权时间并折现得到每条路径的期权价格，取平均即可得到美式期权的定价。

（3）亚式期权

亚式期权的思路和欧式期权一样，但是要模拟路径并取平均值，所以运算量比较大。在我的案例中，对亚式期权每天进行一次算数采样，一年有 365 天。