

# Python程序设计#1作业

班级：2019211308

学号：2019211504

姓名：王乾凯

## 作业题目

每人独立设计并实现一个小型python程序（功能不限），代码需要涉及：`class`类、对象实例化、继承（分别定义父类和子类）、对象方法（`self`参数）、类方法（`@classmethod`）、静态方法（`@staticmethod`）、对象属性、类属性、多态。

## 作业内容

程序源代码嵌入下方的code block中。

```

class fib:
    def __init__(self, a0, a1):
        self.a0 = a0
        self.a1 = a1
        pass
    def cal(n):
        pass
    @classmethod
    def getDefault(cls):
        return cls(0, 1)

class fib_norm(fib):
    def __init__(self, a0, a1):
        super(a0, a1)

    def cal(n):
        for i in range(2, n):
            t = a0
            a0 = a1
            a1 = t
        return a1

class fib_quick(fib):
    def __init__(self, a0, a1):
        super().__init__(a0, a1)
        self.base = [[1, 1], [1, 0]]

    def __matrix_plus(m1, m2):
        res = [[0, 0], [0, 0]]
        for i in range(2):
            for j in range(2):
                res[i][j] = m1[i][j] + m2[i][j]
        return res

    def __matrix_mul(self, m1, m2):
        res = [[0, 0], [0, 0]]
        for i in range(2):
            for j in range(2):
                for k in range(2):
                    res[i][j] += m1[i][k] * m2[k][j];
        return res

    def cal(self, n):
        res = [[1, 0], [0, 1]]
        while(n):
            if (n & 1) :
                res = self.__matrix_mul(res, self.base)
                self.base = self.__matrix_mul(self.base, self.base)
            n = n >> 1
        return res[1][0] * a1 + res[1][1] * a0

```

```

class fib_factory:
    def __init__(self):
        pass

    @staticmethod
    def getInstance(type, a0, a1):
        if type == 'normal':
            return fib_norm(a0, a1)
        elif type == 'quick':
            return fib_quick(a0, a1)

    def getInstance(type):
        if type == 'normal':
            return fib_norm.getDefault()
        elif type == 'quick':
            return fib_quick.getDefault()

if __name__ == '__main__':
    # print('请输入a0和a1:')
    a0 = 1
    a1 = 1
    fib_x = fib_factory.getInstance("quick", a0, a1)
    print(fib_x.cal(10))
    pass

```

## 代码说明

上述代码完成了一个获得斐波那契数列第n个数的算法，实现了两种算法，分别是普通算法和矩阵快速幂算法。

其中fib\_norm为普通 $O(n)$ 算法封装，fib\_quick为 $O(\log n)$ 算法封装，二者都继承于fib父类，使用工厂模式中的静态方法来进行类的实例化。

父类fib定义了类方法getDefault，在不指定斐波那契数列的起点时，定义为0和1。