Task 1: Implement the Sleeping Barber Problem

An older edition of the Tanenbaum textbook describes the "Sleeping Barber" problem and gives a solution in "C". (This material can also be found in a separate file called **SleepingBarberProblem.pdf** in the **lab3** directory.)

Translate this into a working KPL program. You'll also need to create some code to print out what happens when you run the program.

实验设置：

一，没有理发师的时候，客人等待

```
--初始化顾客和理发师
--没有理发师，客人需要等待
for i = 0 to 4
    cThreads[i] = new Thread
    cThreads[i].Init("Customer")
    cThreads[i].Fork(customer, i)
```

结果截图：

```
====================  KPL PROGRAM STARTING  ====================
Executed_NoteInitializing Thread Scheduler...
     (Customer 0) customer into the room
     (Customer 0) if get haircut :FALSE
customer with number<0 >get the wait char.
     (Customer 1) customer into the room
     (Customer 1) if get haircut :FALSE
customer with number<1 >get the wait char.
     (Customer 2) customer into the room
     (Customer 2) if get haircut :FALSE
customer with number<2 >get the wait char.
     (Customer 3) customer into the room
     (Customer 3) if get haircut :FALSE
customer with number<3 >get the wait char.
     (Customer 4) customer into the room
     (Customer 4) if get haircut :FALSE
customer with number<4 >get the wait char.
```

二、理发师被初始化，开始理发

```
--理发师前来，开始工作
bThread = new Thread
bThread.Init("Barber")
bThread.Fork(barber, 1)

WasteTime(1000)
```

结果截图：

```
barber check
barber find there are5 customers waiting for
barber begin haircut
     (Customer 0) Finish Haircut
     (Customer 0) if get haircut :TRUE
     (Customer 5) customer into the room
     (Customer 5) if get haircut :FALSE
customer with number<5 >get the wait char.
barber finish haircut
barber check
barber find there are5 customers waiting for
barber begin haircut
     (Customer 1) Finish Haircut
     (Customer 1) if get haircut :TRUE
```

三、第二批客人加入，没有空位的时候，客人会直接离开

```
--第二批客人进入.没有空位时候，客人直接离开
for i = 5 to 15
    cThreads[i] = new Thread
    cThreads[i].Init("Customer")
    cThreads[i].Fork(customer, i)
endFor

WasteTime(2000)

--加入更多客人，测试系统稳定性
for i = 16 to 30
    cThreads[i] = new Thread
    cThreads[i].Init("Customer")
    cThreads[i].Fork(customer, i)
endFor
```

结果截图：

```
    (Customer 9) customer into the room
    (Customer 9) if get haircut :FALSE
    (Customer 9) customer leave thus the room is full
    (Customer 10) customer into the room
    (Customer 10) if get haircut :FALSE
    (Customer 10) customer leave thus the room is full
```

Task 2: The Gaming Parlor Problem

Here is the problem scenario: groups of customers come in to a "gaming parlor" to play games. They go to the front desk to obtain one or more dice, which are used by the group while they are playing their game, and then returned to the front desk. The front desk is in charge of lending out the dice and collecting them after each game is finished.

结果截图：