begin P\_load arriving //设置八小时内产生load，大于八小时不再产生load

while ac < 8 hr do

begin

wait for stream\_load20 e 1 min

clone 1 load to P\_in

end

send to die

end

begin P\_in arriving //前五名顾客直接分配给五个服务台，后面来的顾客要进行判断哪个队伍最短

set V\_count=V\_count+1 //对顾客进行计数

set A\_count to V\_count

move into conv.sta1

set A\_clockin to ac //ac是仿真钟时刻

if V\_count=1 then send to P\_surve3 //为前五名顾客分配服务台

if V\_count=2 then send to P\_surve2

if V\_count=3 then send to P\_surve4

if V\_count=4 then send to P\_surve1

if V\_count=5 then send to P\_surve5

else send to P\_in1

end

begin P\_in1 arriving //对队伍长度进行判断

choose a queue from among Q1,Q2,Q3,Q4,Q5 whose current loads is minimum save choice as A\_inmin /\* A\_queptr is a load attribute of type QueuePtr \*/

set A\_ac = ac

move into A\_inmin

if A\_inmin is Q1 then

begin

set A\_OL = 1

wait to be ordered on OL(A\_OL)

end

if A\_inmin is Q2 then

begin

set A\_OL = 2

wait to be ordered on OL(A\_OL)

end

if A\_inmin is Q3 then

begin

set A\_OL = 3

wait to be ordered on OL(A\_OL)

end

if A\_inmin is Q4 then

begin

set A\_OL = 4

wait to be ordered on OL(A\_OL)

end

if A\_inmin is Q5 then

begin

set A\_OL = 5

wait to be ordered on OL(A\_OL)

end

end

begin P\_surve1 arriving //服务台1服务胡可

move into Q1

use R1 for streamR20 e 4 min

set A\_clockout to absolute clock

print "custom",A\_count, "wait time was", A\_clockout - A\_clockin, "minutes." to message 输出顾客等待时间到message

tabulate A\_clockout - A\_clockin in T\_time

create 1 load of load type L\_dummy(1) to P\_order //接受完服务的顾客离开系统

clone 1 load to P\_check

move into conv.sta1\_0

travel to conv.sta1\_1

end

begin P\_surve2 arriving //与前面P\_surve1类似

move into Q2

use R2 for streamR20 e 4 min

set A\_clockout to absolute clock

print "custom",A\_count, "wait time was", A\_clockout - A\_clockin, "minutes." to message

tabulate A\_clockout - A\_clockin in T\_time

create 1 load of load type L\_dummy(2) to P\_order

clone 1 load to P\_check

move into conv.sta2\_0

travel to conv.sta2\_1

end

begin P\_surve3 arriving //与前面P\_surve1类似

move into Q3

use R3 for streamR20 e 4 min

set A\_clockout to absolute clock

print "custom",A\_count, "wait time was", A\_clockout - A\_clockin, "minutes." to message

tabulate A\_clockout - A\_clockin in T\_time

create 1 load of load type L\_dummy(3) to P\_order

clone 1 load to P\_check

move into conv.sta3\_0

travel to conv.sta3\_1

end

begin P\_surve4 arriving //与前面P\_surve1类似

move into Q4

use R4 for streamR20 e 4 min

set A\_clockout to absolute clock

print "custom",A\_count, "wait time was", A\_clockout - A\_clockin, "minutes." to message

tabulate A\_clockout - A\_clockin in T\_time

create 1 load of load type L\_dummy(4) to P\_order

clone 1 load to P\_check

move into conv.sta4\_0

travel to conv.sta4\_1

end

begin P\_surve5 arriving //与前面P\_surve1类似

move into Q5

use R5 for streamR20 e 4 min

set A\_clockout to absolute clock

print "custom",A\_count, "wait time was", A\_clockout - A\_clockin, "minutes." to message

tabulate A\_clockout - A\_clockin in T\_time

create 1 load of load type L\_dummy(5) to P\_order

clone 1 load to P\_check

move into conv.sta5\_0

travel to conv.sta5\_1

end

begin P\_order arriving //根据L\_dummy类型将接受服务完成的顾客调离系统

if load type=L\_dummy(1) then

begin

order 1 load from OL(1) to P\_surve1

in case order not filled backorder on OL(1)

end

else if load type=L\_dummy(2) then

begin

order 1 load from OL(2) to P\_surve2

in case order not filled backorder on OL(2)

end

else if load type=L\_dummy(3) then

begin

order 1 load from OL(3) to P\_surve3

in case order not filled backorder on OL(3)

end

else if load type=L\_dummy(4) then

begin

order 1 load from OL(4) to P\_surve4

in case order not filled backorder on OL(4)

end

else if load type=L\_dummy(5) then

begin

order 1 load from OL(5) to P\_surve5

in case order not filled backorder on OL(5)

end

end

begin P\_check arriving //判断哪个队伍的队长最短，并判断是否要换队

choose a queue from among Q1,Q2,Q3,Q4,Q5 whose current loads is minimum save choice as A\_outmin //队长最短的队伍赋值给A\_outmin

choose a queue from among Q1,Q2,Q3,Q4,Q5 whose current loads is maximum save choice as A\_outmax //队长最长的队伍赋值给A\_outmax

choose a load from among A\_outmax load list whose A\_ac is maximum save choice as V\_move //从队长最长的队伍中选一个load赋值给V\_move

if A\_outmax current loads-A\_outmin current loads > 1 then

begin

print OL(V\_move A\_OL) "中" ,V\_move, "开始换队" to message

order load V\_move from OL(V\_move A\_OL) to P\_in1

end

else send to die

end