

```

type pickup_chess = MONITOR {
    bool white_turn = true;
    semaphore s_white, s_black;
    int s_white_count, s_black_count;
    InterfaceModule IM;
    DEFINE white(), black();
    USE enter(), leave(), wait(), signal();

```

```

void white(){
    enter(IM);
    if (!white_turn)
        wait(s_white, s_white_count ,IM);
    white_turn = false;
    {拣白子};
    signal(s_black, s_black_count, IM);
    leave(IM);
}

```

```

void black(){
    enter(IM);
    if (white_turn)
        wait(s_black, s_black_count ,IM);
    white_turn = true;
    {拣黑子};
    signal(s_white, s_white_count, IM);
    leave(IM);
}

```

cobegin

```

Process P1(){
    pickup_chess.white();
    othres;
}

```

```

Process P2(){
    pickup_chess.black();
    othres;
}

```

coend