

Waiter

boolean end = false, startedPreparation = false;

```
while(!end){
    if (Table.studentHasEntered()){ // se um deles entrou
        Table.saluteTheClient(); //empregado diz ola
        while(!Table.readTheMenu()){
            Table.presentMenu();
        }

    }
    else if (Table.studentHasLeft()){ // se um deles sair
        Table.sayGoodbye(); //o empregado diz goodbye
    }
    else if (Table.allHaveLeft()) { // se eles saíram do restaurante
        end = true;

    } else if (Bar.callTheWaiter()) { //estudante chama empregado
        Bar.getThePad();
        while (!Table.describeTheOrder()){ // tira o pedido ate o estudante acabar
            Table.takeTheOrder(); // empregado vai buscar o pedido
        }
        while (!Kit.startPreparation()){
            Kit.handTheNoteToTheChef(); // dar a ordem ao chef quanto o chefe
            // ainda nao comecou a preparar
        }
        startedPreparation = true;
    } else if (Table.shouldHaveArrivedEarlier()) { // o estudante diz olha quero pagar
        Bar.prepareTheBill();
        while(!Table.honorTheBill()){ //o empregado fica aqui ate ele pagar
            Table.presentTheBill();
        }
    } else if (Bar.signalTheWaiter()) {startedPreparation = true;}
    if (startedPreparation){ // o chef comecou a preparar pratos

        while(!Kit.haveAllPortionsBeenDelivered()){ // enquanto nao tiver preparado
            //todas as porcoes
            if (Kit.alertTheWaiter()) { // o chef diz para ir buscar comida
                Kit.collectPortion(); //coleta a comida
                Table.deliverPortion();
            }
            startedPreparation = false;
        }
    }
    bar.lookAround();
}
```

Student

```
boolean first = false, organizingOrder = true, allHaveChosen = false, lastToEat= false, orderDone = true;
```

```
int m = 0;
```

```
GOING_TO_THE_RESTAURANT();
```

```
Table.TAKING_A_SEAT_AT_THE_TABLE(); // set first to true if first
```

```
while(!Table.saluteTheClient()){  
    Table.studentHasEntered();  
}
```

```
Table.SELECTING_THE_COURSES();
```

```
if(!first){  
    Table.informCompanions(); // imformar o 1 que já tem o pedido e dizer lhe  
}
```

```
do{
```

```
    if (first){  
        while (organizingOrder){  
            Table.prepareOrder(); //anota o pedido dos companheiros  
            if (allHaveChosen) {  
                organizingOrder = false;  
                while(!Bar.getThePad()){ //enquanto o empregado nao vier com papel  
                    Bar.callTheWaiter(); // chama o  
                }  
                Table.describeOrder();  
                m = x courses;  
            }  
        }  
    }  
}
```

```
if(receivedCourse){  
    table.enjoyMeal();  
    receivedCourse = false;  
    if (lastToEat) {  
        Bar.signalTheWaiter(); // pede para trazer o proximo prato  
        m--;  
    }  
}
```

```
Table.CHATTING_WITH_COMPANIONS();  
}while(m>0);
```

```
if (Table.shouldHaveArrivedEarlier()){  
    while(!Table.presentTheBill()) {//pagamento - ultimo e que paga  
        Table.honorTheBill();  
    }  
}
```

```
}
```

```
Table.going_home();
```