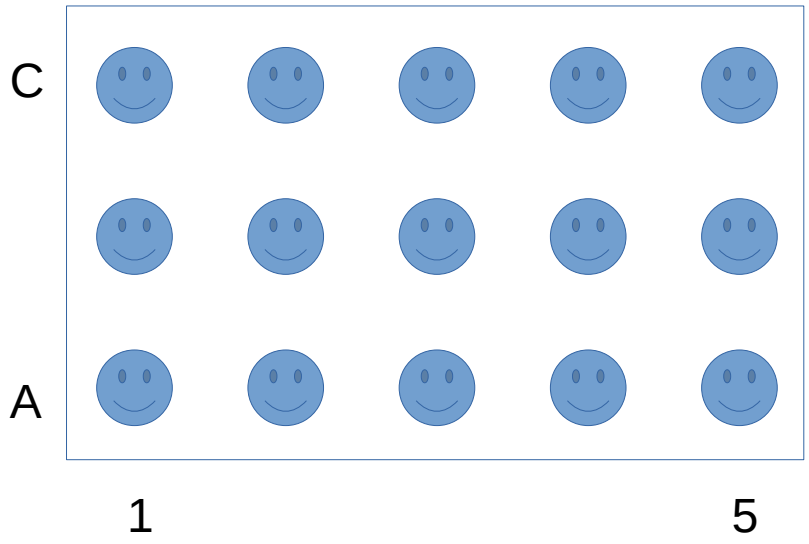
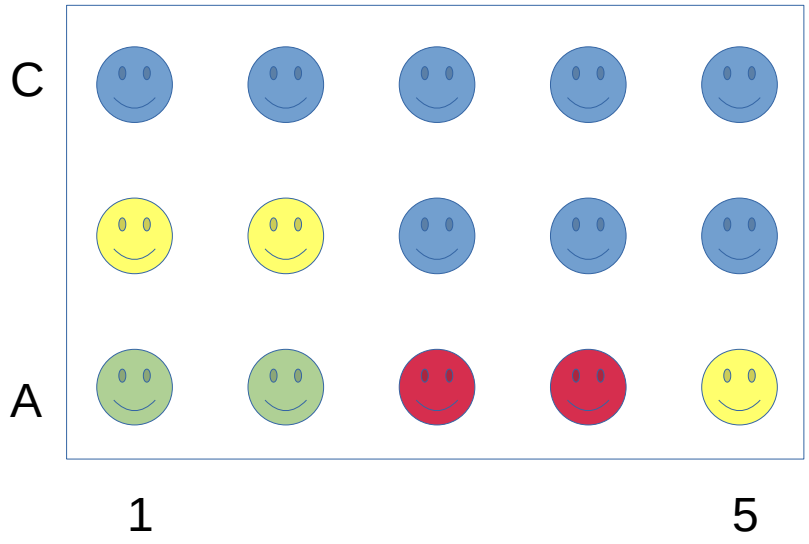
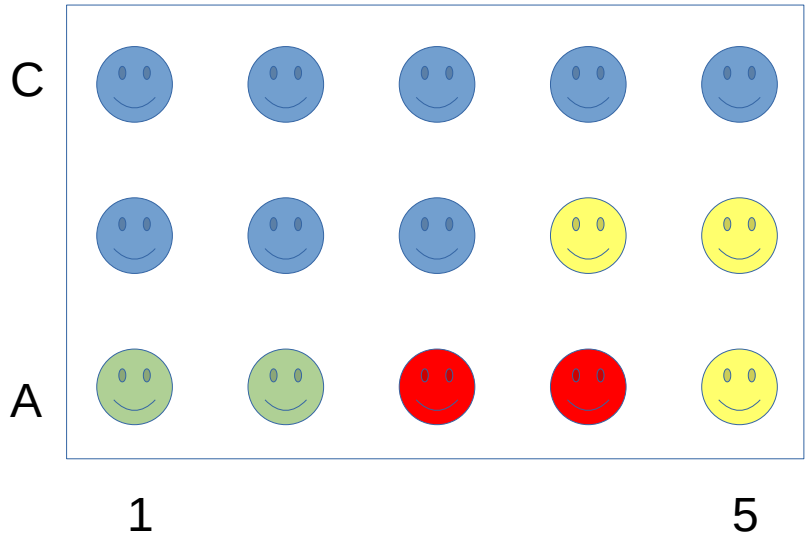
Initial planning



Empty cinema



Option 1: Fill By Cinema A1-A5, B1 – B5, C1 -C5 – but people could be on opposite ends of the cinema



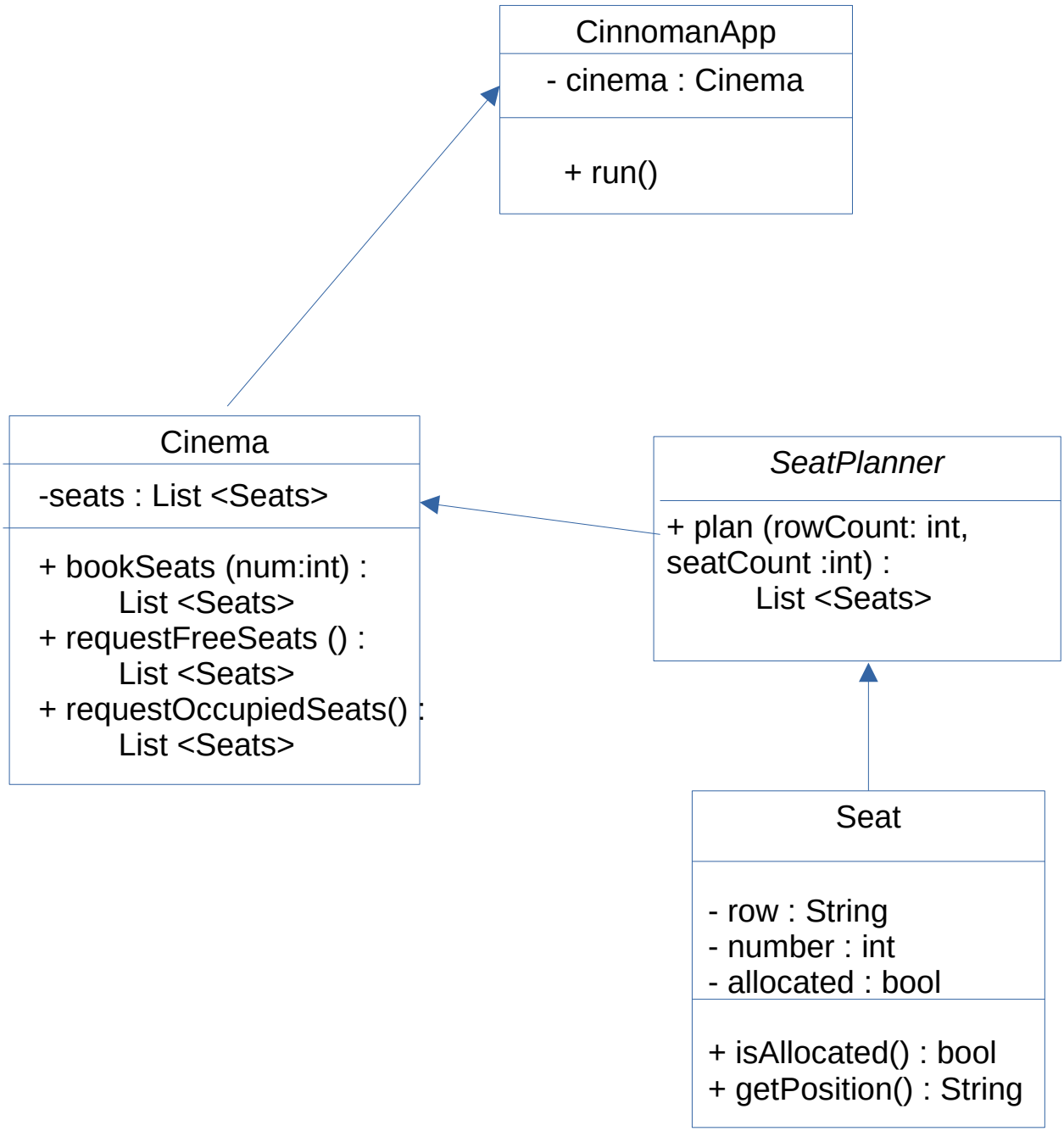
Option 2: Fill By Cinema A1-A5, B5 – B1, C1 -C5 – enables people to sit close together

Option 3. Fill By Row But what happens in this situation? Do we continue to fill from the front or just stop since we've not got 3 spaces left on the same row?

I think option 2 makes the most sense to fill the maximum number of seats and keep customers close together, however it doesn't allow customers to select their own seats

Possible ways to proceed:
1. A pre-populated list of seats could be pushed to a queue at the start and popped off as and when requested, however this may break the requirement for the seats to be recorded as allocated
2. An ordered list of seats is created at the start, and each seat is marked as unoccupied. Then as seats are requested the seats are marked as allocated and returned to the requester
3. A 2 Dimensional array could be initialised at the start, and then as seats are requested the array is indexed and the appropriate seats are returned and marked as allocated – bit messy that though

So we need a cinema, populated with seats, a ticket request with 1 to 3 seats, list of tickets allocated, list of tickets free, and a console application with random number generator


CinnomanApp
- cinema : Cinema
+ run()
Cinema
-seats : List <Seats>
+ bookSeats (num:int) :
List <Seats>
+ requestFreeSeats () :
List <Seats>
+ requestOccupiedSeats() :
List <Seats>
SeatPlanner
+ plan (rowCount: int,
seatCount :int) :
List <Seats>
Seat
- row : String
- number : int
- allocated : bool
+ isAllocated() : bool
+ getPosition() : String