```
function ticket = generateTambolaTicket()
% Main controller function

while true
    colCount = generateColumnCounts();
    mask = generateStructure(colCount);

    if ~validateStructure(mask)
        continue
    end

    ticket = fillTicket(mask, colCount);

    if validateTicket(ticket)
        return
    end
end

end

function colCount = generateColumnCounts()
% Decide how many numbers per column (sum = 15, each 1–3)

colCount = ones(1,9);
remaining = 6;

while remaining > 0
    idx = randi(9);
    if colCount(idx) < 3
        colCount(idx) = colCount(idx) + 1;
        remaining = remaining - 1;
    end
end

end

function mask = generateStructure(colCount)
% Generate 3x9 logical structure (1 = number, 0 = blank)

mask = false(3,9);
rowCount = zeros(1,3);

for col = 1:9
    rows = randperm(3, colCount(col));
    for r = rows
        if rowCount(r) < 5
            mask(r,col) = true;
            rowCount(r) = rowCount(r) + 1;
        end
    end
end
```

```
end
function isValid = validateStructure(mask)
% Validate structural constraints

isValid = true;

if sum(mask(:)) ~= 15
    isValid = false; return
end

if any(sum(mask,2) ~= 5)
    isValid = false; return
end

colCount = sum(mask,1);
if any(colCount < 1 | colCount > 3)
    isValid = false; return
end

end
function range = getColumnRange(colIndex)
% Return valid numeric range for column

if colIndex == 1
    range = 1:9;
elseif colIndex == 9
    range = 80:90;
else
    range = (colIndex-1)*10 : (colIndex*10 - 1);
end

end
function values = generateColumnNumbers(colIndex, count)
% Generate sorted numbers for one column

range = getColumnRange(colIndex);
values = sort(randsample(range, count));

end

function ticket = fillTicket(mask, colCount)
% Fill structure with actual numbers

ticket = zeros(3,9);
usedNumbers = [];

for col = 1:9
    rows = find(mask(:,col));
    if isempty(rows)
        continue
    end

    range = setdiff(getColumnRange(col), usedNumbers);
    values = sort(randsample(range, length(rows)));
    for k = 1:length(rows)
        ticket(rows(k), col) = values(k);
    end

    usedNumbers = [usedNumbers values];
end

end

function isValid = validateTicket(ticket)
% Final validation of Tambola ticket

isValid = true;

% Row constraint
if any(sum(ticket > 0,2) ~= 5)
    isValid = false; return
end

% Column constraint
colCount = sum(ticket > 0,1);
if any(colCount < 1 | colCount > 3)
    isValid = false; return
end

% Range + uniqueness
nums = ticket(ticket > 0);
if length(nums) ~= length(unique(nums))
    isValid = false; return
end

% Column ordering
for c = 1:9
    vals = ticket(:,c);
    vals = vals(vals > 0);
    if any(diff(vals) < 0)
        isValid = false; return
    end
end

end
```

```
ans =

     5    16    20     0    41    52     0     0     0
     6    17    21     0    43     0     0    71     0
     0     0    26    30     0     0    67    72    87
```

Published with MATLAB® R2020b