手写版：

1.a

```
public DownloadInfo getDownloadInfo(String title) {
    for (DownloadInfo info : downloadList) {
        if (info.getTitle().equals(title)) {
            return info;
        }
    }
    return null;
}
```

1.b

```
public void updateDownloads(List<String> titles) {
    for (String title : titles) {
        DownloadInfo foundInfo = getDownloadInfo(title);
        if (foundInfo == null) {
            downloadList.add(new DownloadInfo(title));
        }
        else {
            foundInfo.incrementTimesDownloaded();
        }
    }
```

```
}
```

2.a

```
public TokenPass(int playerCount) {
    board = new int[playerCount];
    for (int i = 0; i < playerCount; i++)
        board[i] = 1 + (int) (10 * Math.random());
    }
    currentPlayer = (int) (playerCount * Math.random());
}
```

2.b

```
public void distributeCurrentPlayerTokens() {
    int nextPlayer = currentPlayer;
    int numToDistribute = board[currentPlayer];
    board[currentPlayer] = 0;
    while (numToDistribute > 0) {
        nextPlayer = (nextPlayer + 1) % board.length;
        board[nextPlayer]++;
        numToDistribute--;
    }
}
```

4.a

```
public skyView(int numRows, int numCols, double[] scanned) {
    view = new double[numRows][numCols];
    int i = 0;
    for (int row = 0; row < numRows; row++) {
        if (row % 2 == 0) {
            for (int col = 0; col < numCols; col++) {
                view[row][col] = scanned[i];
                i++;
            }
        } else {
            for (int col = numCols - 1; col >= 0; col--) {
                view[row][col] = scanned[i];
                i++;
            }
        }
    }
}
```

4b.

```
public double getAverage(int startRow, int endRow, int startCol, int
endCol) {
    double sum = 0.0;
    int count = 0;
    for (int row = startRow; row <= endRow; row++) {
        for (int col = startCol; col <= endCol; col++) {
            sum += view[row][col];
            count++;
        }
    }
    return sum / count;
}
```

手写版：

1.a

```
public DownloadInfo getDownloadInfo(String title) {
    for (DownloadInfo info : downloadList) {
        if (info.getTitle().equals(title)) {
            return info;
        }
    }
    return null;
}
```

1.b

```
public void updateDownloads(List<String> titles) {
    for (String title : titles) {
        DownloadInfo foundInfo = getDownloadInfo(title);
        if (foundInfo == null) {
            downloadList.add(new DownloadInfo(title));
        }
        else {
            foundInfo.incrementTimesDownloaded();
        }
    }
```

```
}
```

2.a

```
public TokenPass(int playerCount) {
    board = new int[playerCount];
    for (int i = 0; i < playerCount; i++)
        board[i] = 1 + (int) (10 * Math.random());
    }
    currentPlayer = (int) (playerCount * Math.random());
}
```

2.b

```
public void distributeCurrentPlayerTokens() {
    int nextPlayer = currentPlayer;
    int numToDistribute = board[currentPlayer];
    board[currentPlayer] = 0;
    while (numToDistribute > 0) {
        nextPlayer = (nextPlayer + 1) % board.length;
        board[nextPlayer]++;
        numToDistribute--;
    }
}
```

4.a

```
public skyView(int numRows, int numCols, double[] scanned) {
    view = new double[numRows][numCols];
    int i = 0;
    for (int row = 0; row < numRows; row++) {
        if (row % 2 == 0) {
            for (int col = 0; col < numCols; col++) {
                view[row][col] = scanned[i];
                i++;
            }
        } else {
            for (int col = numCols - 1; col >= 0; col--) {
                view[row][col] = scanned[i];
                i++;
            }
        }
    }
}
```

4b.

```
public double getAverage(int startRow, int endRow, int startCol, int
endCol) {
    double sum = 0.0;
    int count = 0;
    for (int row = startRow; row <= endRow; row++) {
        for (int col = startCol; col <= endCol; col++) {
            sum += view[row][col];
            count++;
        }
    }
    return sum / count;
}
```