

parseEntryPoint(line: String; entryPoints: Map < String, EntryPoint >; coordChecker: List < Coord >; referencesMadeByEntryPoints: HashSet < String >)

String[] parts = line.split("\\s+")

parts.length != ENTRY\_POINT\_PARTS\_COUNT

T

F

throw new IllegalArgumentException(ERROR\_INVALID\_ENTRY\_POINT\_FORMAT + line)

Ø

String name = parts[0]

name.length() > ENTRY\_POINT\_NAME\_MAX\_LENGTH

T

F

throw new IllegalArgumentException(ERROR\_ENTRY\_POINT\_NAME\_TOO\_LONG + name)

Ø

entryPoints.containsKey(name)

T

F

throw new IllegalArgumentException(ERROR\_DUPLICATE\_ENTRY\_POINT\_NAME + name)

Ø

double x = checkCoordinateComponent(parts[ENTRY\_POINT\_X\_INDEX])

double y = checkCoordinateComponent(parts[ENTRY\_POINT\_Y\_INDEX])

Coord epCoord = new Coord(x, y)

checkWhetherCoordsFarEnough(coordChecker, epCoord)

String destination = parts[ENTRY\_POINT\_DEST\_INDEX]

referencesMadeByEntryPoints.add(destination)

int freq = Integer.parseInt(parts[ENTRY\_POINT\_FREQ\_INDEX])

EntryPoint ep = new EntryPoint()

ep.coord = new Coord(x, y)

ep.freq = freq

ep.intersectionName = destination

entryPoints.put(name, ep)