1603. Design Parking System

Solved

Easy Topics Companies Hint

Design a parking system for a parking lot. The parking lot has three kinds of parking spaces: big, medium, and small, with a fixed number of slots for each size.

Implement the `ParkingSystem` class:

- `ParkingSystem(int big, int medium, int small)` Initializes object of the `ParkingSystem` class. The number of slots for each parking space are given as part of the constructor.
- `bool addCar(int carType)` Checks whether there is a parking space of `carType` for the car that wants to get into the parking lot. `carType` can be of three kinds: big, medium, or small, which are represented by `1`, `2`, and `3` respectively. **A car can only park in a parking space of its** `carType`. If there is no space available, return `false`, else park the car in that size space and return `true`.

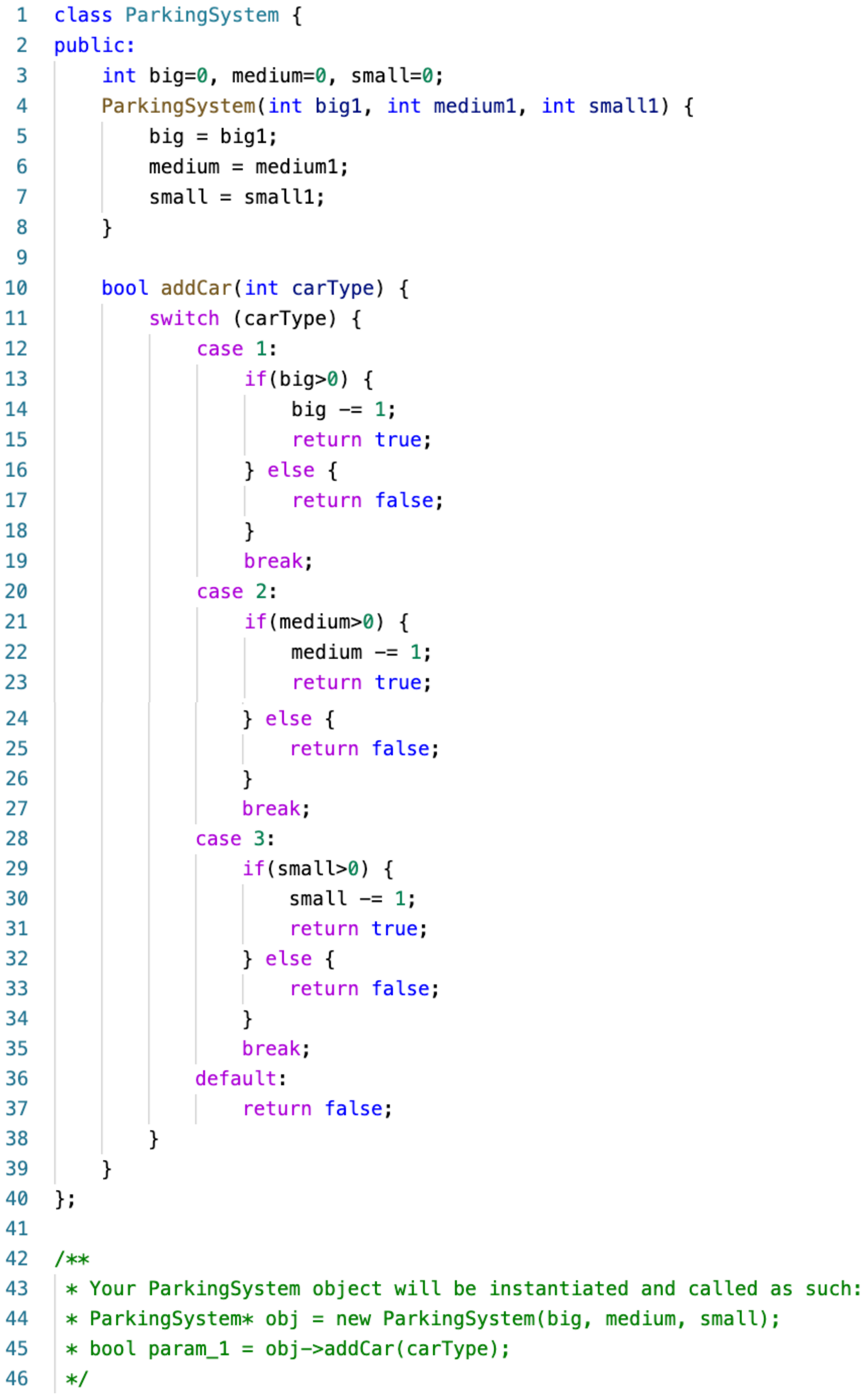
Example 1:

```
Input
["ParkingSystem", "addCar", "addCar", "addCar", "addCar"]
[[1, 1, 0], [1], [2], [3], [1]]
Output
[null, true, true, false, false]

Explanation
ParkingSystem parkingSystem = new ParkingSystem(1, 1, 0);
parkingSystem.addCar(1); // return true because there is 1 available slot
for a big car
parkingSystem.addCar(2); // return true because there is 1 available slot
for a medium car
parkingSystem.addCar(3); // return false because there is no available slot
for a small car
parkingSystem.addCar(1); // return false because there is no available slot
for a big car. It is already occupied.
```

Constraints:

- `0 <= big, medium, small <= 1000`
- `carType` is `1`, `2`, or `3`
- At most `1000` calls will be made to `addCar`



```cpp
class ParkingSystem {
public:
    int big=0, medium=0, small=0;
    ParkingSystem(int big1, int medium1, int small1) {
        big = big1;
        medium = medium1;
        small = small1;
    }

    bool addCar(int carType) {
        switch (carType) {
            case 1:
                if(big>0) {
                    big -= 1;
                    return true;
                } else {
                    return false;
                }
                break;
            case 2:
                if(medium>0) {
                    medium -= 1;
                    return true;
                } else {
                    return false;
                }
                break;
            case 3:
                if(small>0) {
                    small -= 1;
                    return true;
                } else {
                    return false;
                }
                break;
            default:
                return false;
        }
    }
};

/**
 * Your ParkingSystem object will be instantiated and called as such:
 * ParkingSystem* obj = new ParkingSystem(big, medium, small);
 * bool param_1 = obj->addCar(carType);
 */
```