

[프로그래머스]주차 요금 계산

▼ 상태	진행 중
📅 날짜	@2022년 6월 3일
≡ 공부유형	스터디
≡ 알고리즘	
▼ 사이트	
☑ 깃허브	☑

코딩테스트 연습 - 주차 요금 계산

문제 설명 주차장의 요금표와 차량이 들어오고(입차) 나간(출차) 기록이 주어졌을 때, 차량별로 주차 요금을 계산하려고 합니다. 아래는 하나의 예시를 나타냅니다. 어떤 차량이 입차된 후에 출차된 내역이 없다면, 23:59에 출차된 것으로 간주합니다. 0000번 차량은 18:59에 입차된 이후, 출차된 내역이 없습니다. 따라서, 23:59에 출차된 것으로 간주

🔗 <https://programmers.co.kr/learn/courses/30/lessons/92341?language=java#>



개념

- hashmap 정렬 (문자열 사전순으로 키 정렬)
 - `Map<String, Integer> sortedmap = new TreeMap<>(map);`
- 문자열 → 숫자
 - `Integer.parseInt(str)`
 - `Integer.valueOf(str)`
 - `parseInt()` 와 가장 큰 차이점은, `valueOf()` 메소드는 문자열을 변환하여 Integer Object를 리턴
 - `parseInt()` 메소드는 primitive type인 int를 리턴
 - `int intValue1 = Integer.valueOf(str1).intValue();`

풀이

1. 누적주차시간 계산

- 나간 시간 - 들어온 시간
- 여러번 있을 경우 or 들어온 시간만있는 경우 어떻게 할 것인가
- hashmap에 차번호로 값을 저장할거?

2. 주차요금 계산

기본 시간(분) 기본 요금(원) 단위 시간(분) 단위 요금(원)

- 생각할 것
 1. 입차 후 출차가 없는 차량은 따로 23:59에서 입차시간을 빼줘야함.
 2. 입차 출차 1번씩 하고 그 뒤에 또 들어온 차량은 누적해서 시간을 계산해야함.
- 두개의 hashmap 사용
 1. `parking` : 입차한 차들 기록 관리(차량번호, 입차시간(분)) = 현재 주차중인 차
 2. `fee` : 출차한 차들 기록 관리 (차량번호, 차량의 총 주차 시간(분))

- a. 출차하며 parking에서 해당 차량 제거하여 문제1 해결
- b. 총 주차시간 누적하여 더할 수 있어 문제2 해결

코드

```
import java.util.*;
//1. 누적주차시간 계산(분으로)
//2. 주차요금 계산

public class 주차요금계산 {

    public static void main(String[] args) {
        int[] n = {180, 5000, 10, 600};
        String[] n2 = {"05:34 5961 IN", "06:00 0000 IN", "06:34 0000 OUT", "07:59 5961 OUT", "07:59 0148 IN", "18:59 0000 IN", "19:09 0148 OUT"};

        System.out.println(solution(n, n2));
    }

    public static int[] solution(int[] fees, String[] records) {
        int[] answer; // 차량번호가 작은 자동차부터 주차요금을 정수 배열에 담아 return

        //입력값으로부터 주차한 차들 정보가진 리스트 parking
        HashMap<String, Integer> parking = new HashMap<>(); // Integer[]

        //나간 차들 요금 계산 위해 ( 한번 들어온차 말고 추후에 또 들어왔을때 한꺼번에 합쳐서 요금 청구 위해)
        //PriorityQueue<car> fee = new PriorityQueue<>();
        //pq는 자동 정렬은 되지만 있는지 확인하고 빼는게 힘들어
        HashMap<String, Integer> fee = new HashMap<>(); // Integer[]

        for (String r : records) { // "05:34 5961 IN"
            String[] record = r.split(" "); // 띄어쓰기 기준으로 항목별로 배열에 넣음
            int t = getMin(record[0]); // 시간
            String n = record[1]; // 차량번호
            String io = record[2]; //출입

            // IN 이면
            if(io.equals("IN")) {
                //차량번호를 key로 하고, value는 들어온 시간을 분으로 바꾼 것.
                parking.put(n, t);
            }

            // OUT 이면
            }else {
                if(!fee.containsKey(n)) { //이미 한번 들어왔던 차가 아니라면
                    fee.put(n, t - parking.get(n)); // 요금 청구할 시간 저장
                }else { // 한번입출차하여 누적시간이 있다면
                    fee.put(n, fee.get(n) + t - parking.get(n)); //기존 시간에 추가해서 더하기
                }
                parking.remove(n);
            }
        }

        //parking에 남아있는 출차하지 않은 차 요금 계산하기
        for (String key : parking.keySet()) { //key = 차량번호
            if(!fee.containsKey(key)) {
                fee.put(key, (23*60+59)-parking.get(key));
            }else {
                fee.put(key, (23*60+59)-parking.get(key)+fee.get(key));
            }
        }

        //fee 차량번호순으로 정렬하기
        Map<String, Integer> sortedFee = new TreeMap<>(fee);

        //fee에 저장되어있는 것들 요금 계산해서 출력하기
        //fees[기본 시간(분) 기본 요금(원) 단위 시간(분) 단위 요금(원)]
        answer = new int[fee.size()];
        int i = 0;
        for (String car : sortedFee.keySet()) { //key = 차량번호
            int min = sortedFee.get(car);
            //요금 = 기본요금 + ((min-기본시간)/단위시간) *단위요금
            if(min<=fees[0]) {
                answer[i++] = fees[1];
                System.out.println(i+ " "+car+ "요금 : "+ fees[1]);
            }else {
                answer[i++] = (int) (fees[1] + Math.ceil(((double)(min-fees[0])/fees[2]))*fees[3]);
                System.out.println(i+ " "+car+ "요금 : "+ (int) (fees[1] + Math.ceil(((double)(min-fees[0])/fees[2]))*fees[3]));
            }
        }

        return answer;
    }
}
```

```

public static int getMin(String time) { // "05:34" 형태의 문자열을 int형 분으로 변환.
    String[] t = time.split(":");
    return Integer.parseInt(t[0])*60 + Integer.parseInt(t[1]);
}

}

```

참고

▼ HashMap 키로 정렬하기

[Java]HashMap 키(Key) 정렬 방법

이번 포스팅에서는 Java에서 HashMap 키를 기준으로 정렬하는 방법을 소개합니다. HashMap은 순서를 보장하지 않습니다. 아래 간단한 HashMap 예제를 보면 삽입 순서대로 출력되지 않는 것을 확인할 수 있습니다. Map

```
map = new HashMap (); map.put("John", 34); map.put("Jane", 26); map.put("Tom", 27); map.put("Bill",
```

🔗 <https://developer-talk.tistory.com/395>



▼ 문자열 ↔ 숫자

[Java] String을 int로, int를 String으로 변환하기 (문자열 숫자 변환)

String 문자열을 int로 변환하기 위해서는 java.lang.Integer 클래스의 parseInt()와 valueOf() 메소드를 사용할 수 있습니다. static int parseInt(String s) java.lang.Integer 클래스의 static 메소드인 parseInt() 메소드는 파라미터로 숫자로 변환할 문자열을 입력받고, 입력받은 문자열을 integer로 변환한 int 값을 리턴합니다.

🔗 <https://hianna.tistory.com/524>

