|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 객체지향 시스템 설계 | | | | | |
| 학번 | 2018023390 | **이름** | 이서연 | **제출일** | 2021.05.01 |

1. Schedule class
   1. [Copy Constructor] Schedule(String name, Schedule obj)

Schedule obj와, 새로운 schedule의 이름을 인풋으로 받아 obj의 내용을 그대로 복사한다. array 형태의 plan의 경우 deep copy를 위해 for loop을 돌며, obj의 내용을 그대로 복사한다. 한번 실행될 때마다 static variable scheduleNum이 1씩 늘어난다.

* 1. [Constructor] Schedule(String name, int days)

이름과 days를 input으로 받는 constructor이다.

* 1. [getters] getName(), getDays(), getExpense(),

name, days, expense를 리턴하는 getter method이다.

* 1. [getters] getActivity(int day, int time)

plan을 그대로 리턴 하는 것은 위험할 것 같 것 같다고 판단했고, 굳이 plan을 리턴 할 필요가 없을 것 같아 day와 time이 주어지면, 그 시간대의activity를 리턴하는 형태로 getter를 만들었다.

* 1. [setter] setActivity(int day, int time, Activity obj)

day와 time을 넣어주면, plan에서 그 날짜, 시간대에 activity를 넣는 method이다.

* 1. [method] Boolean checkPlan(int day, int time)

plan에서 해당 시간, 해당 날짜에 Plan이 있는지 확인한다.

* 1. [method] boolean checkActivity(Activity obj)

전체 plan에서 obj와 동일한 activity가 있는지 체크하는 method로, 12까지, 이 schedule의 전체 day까지 for loop을 돌며 checkPlan을 실행해 activity가 존재하는지 확인하고, 만약에 존재한다면 obj가 그 시간, 그 날짜의 activity와 동일한지 확인한다.

* 1. [method] void printPlan()

먼저 16\*6으로 “-“를 출력해 구분선을 만들고, 0~12까지(t) , 또 1~5까지 for loop을 돌면서 정보를 출력하게 된다. t ==0 일 때는 Day 정보를 출력하게 되고, 그 이후로는 getActivity(1-D) method와 Activity class의 getName method를 통해 Activity의 이름을 불러온다. 이후 구분선을 한번 더 출력하고, total expense를 출력한다.

* 1. [method] void delActivity(int day, int time)

입력받은 day, time을 토대로 그 plan에 있는 activity 를 null로 바꾼다. activity를 삭제하면, 그 activity의 price만큼 expense를 깎기 위해 plan에서 그 시간, 그 날짜 대의 activity에서 Activity class의 getPrice method를 호출해 차감한다.

1. Activity class
   1. [Copy constructor] Activity(Activity obj)

Activity obj를 input으로 넣어주면, name, location, price를 복사해 넣는다.

* 1. [constructor] Activity(String name, String location, int price)

name, location, price를 input으로 넣고 상응하는 Activity를 생성한다.

* 1. [method] boolean equals(Activity obj)

obj의 name, location, price를 비교해 일치하면 true를 반환하고, 다르면 false를 반환한다.

* 1. [getters] getName(), getPrice()

name, price를 반환하는 getter method

* 1. [method] String toString()
     1. 예를들어 hiking method 라면 “Hiking(Mountain, 0 won)” 형식으로, [name] + “(“ +. [place] +”,” + [price] + “)” 가 출력되도록 했다.

1. TravelScheduler-main
   1. 기본 설정

ScheduleList와 activityList는 갯수가 각각 5, 8로 고정되어 있으며, 이에 맞게 생성한다. activityList는 기본 설정이 정해져 있기에 이에 맞춰 초기 설정을 하였다.

* 1. while loop

메뉴를 출력하고, s1 변수에 입력을 받는다.

* + 1. 1번 메뉴 (Select schedule)을 선택할 경우
       1. schedule 리스트를 출력하는데, 리스트의 길이(=5) 만큼 forloop을 돌면서 scheduleNum보다 작으면 schedule의 이름을 출력하고(getName) 크거나 같으면 EMPTY SCHEDULE을 출력한다.
       2. s2 변수에 선택할 schedule 이름을 받는데, 0을 입력받을 경우 처음 반복문으로 돌아가며, EMPTY SCHEDULE 에 해당하는, scheduleNum보다 큰 값을 입력받아도 처음으로 돌아간다.
       3. scheduleNum 미만의, 제대로 된 schedule을 선택했을 경우, 메뉴를 출력하고 s3 변수에 입력을 받는다.
          1. 1) Add activity 를 선택했을 경우

activityList를 forloop을 돌면서 출력한다.

activity, 날짜, 시간을 입력받아 tmp\_num 1, 2, 3 에 입력을 받는다.

Schedule 클래스의 checkPlan, checkActivity 메소드를 이용해 그 시간, 그날짜에 activity를 넣을 수 있는지, activity 가 겹치는 것이 있는지 확인하고 가능할 경우 setActivity method를 통해 activity를 추가한다. 실패할 경우 “Fail to add activity”를 출력한다.

* + - * 1. 2)Remove activity를 선택했을 경우

먼저 Schedule 클래스의 printPlan() 메소드를 실행해 plan을 출력하고, delActivity 함수를 이용해 그 날짜, 그 시간에 맞는 activity를 제거한다. 제거에 성공하면 “Removed successfully” 가 출력된다.

* + - * 1. 3) Print schedule을 선택할 경우

schedule List의 printPlan메소드를 실행한다.

* + 1. 2번 메뉴 (Edit schedule)을 선택할 경우
       1. 메뉴를 출력하고 s2에 입력을 받는다.
          1. 0을 입력했을 경우

처음부터 다시 시작한다.

* + - * 1. 1)Make a new schedule 을 선택했을 경우

새로 생성할 스케줄의 이름과, 여행할 일자를 입력받고 새 스케줄을 입력받아 Schedule List에 추가한다.

* + - * 1. 2)Copy an existing schedule를 선택했을 경우

ScheduleList를 출력하고, 복사할 스케줄과, 새로운 스케줄의 이름을 입력받는다. 이를 토대로 새로운 Schedule 객체를 Copyconstructor로 생성해 Schedule List에 추가한다.

* + 1. 3번 메뉴 (End schedule)을 선택할 경우

exit(0)을 통해 프로그램을 종료시킨다.

1. 추가
   1. 추가적으로, 날짜나 시간의 입력을 받아야 하는 상황에서, 본인이 테스트 하는 과정에서 range를 넘겨서 입력을 넣어주는 실수를 계속 하는 관계로 range 안에서만 입력을 받고, 넘겨서 받을 경우 처음으로 돌아가도록 구현했다.