|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 객체지향 시스템 설계 | | | | | |
| 학번 | 2018023390 | **이름** | 이서연 | **제출일** | 2021.05.17 |

1. **Class Activity**
   1. instance 정보 : name, location, price
   2. constructor 정보
      1. public Activity(Activity obj) : copy constructor
      2. public Activity(String name, String location, int price) : name, location, price를 받아 activity object를 생성함.
   3. method 정보
      1. public boolean equals(Activity obj) : 이름, location, price를 받아 비교함.
      2. public String toString() : name + “(“ + location + “, “ + price “ won” 으로 출력함.
      3. public String getName(), public int getPrice() : int와 price를 반환하는 getter
      4. public int getActualPrice(Person person) : Price를 반환함.
2. **Class ExtremeActivity extends Activity**
   1. instance 정보 : minHeight, minWeight
   2. constructor 정보 : (1) Copy constructor, (2) name, location, price, minHeight, minWeight를 받아 생성하는 생성자.
   3. method정보
      1. getMinHeight(), getMinWeight() : minheight, minweight 를 반환하는 getter
      2. public int getActuatlPrice(Person person) : person의 나이가 60 이상일 경우 0.7을 곱해 반환함. (30% 할증) 아니면 그대로 반환함.
3. **Class ShowActivity extends Activity**
   1. instance 정보 : minAge
   2. constructor 정보 : (1) copy constructor, (2) name, location, price, minAge를 받아 생성하는 생성자
   3. method 정보
      1. getMinAge() : age 반환하는 getter
      2. public int getActualPrice(Person person) : 나이가 20세 이하일 경우 0.8을 곱해 반환함.(20% 할증) 아니면 그대로 반환함.
4. **Class Person**
   1. instance 정보 : name, age, height, weight
   2. constructor 정보 : name, age, height, weight 를 받아 생성하는 생성자
   3. method 정보
      1. getAge(), getHeight, getWeight() : age, height, weight 를 반환하는 getter
      2. toString() : Name, Age, Height, Weight 정보를 출력함.
5. **Class Schedule**
   1. instance 정보 : name, days, Activity[][] plan, expense, Person[] member, (static) scheduleNum
   2. constructor 정보 : (1) name, Schedule obj를 받는 copy constructor, (2) name, days, Person[] member 를 받는 constructor
   3. method 정보
      1. getName(), getDays(), getExpense(), getMember() : getter
      2. public boolean checkPlan(int day, int time) : 그 시간, 그 날짜의 activity 반환
      3. public Activity getActivity(int day, int time) : showActivity인지, extremeActivity인지, Activity인지에 따라서 새로운 Activity를 만들어서 반환함.
      4. public void setActivity(int day, int time, Activity obj) : activity 종류에 따라 만들어서 plan에다가 추가함.
      5. public boolean checkActivity(Activity obj) : obj 가 plan에 동일한게 있는지 확인.
      6. printPlan, delActivity : 1번과제와 동일하나, getActualPrice를 사용함.
6. **Exception Handling**
   1. Programmer-defined Exceptions (try 의 위치)
      1. InvalidAccessException
         1. 메뉴 선택 : line 79 , 110, 163
            1. 정해진 메뉴 안에서 고르지 않았을 때 exception 발생
         2. 이미 그 시간대에 activity가 존재함 : 314
            1. checkPlan method를 통해 그 시간, 그 날짜에 activity가 있는지 확인하고 있으면 exception을 발생시킴.
         3. 제거하려는 activity가 없을때, 수정하려는 schedule이 없을 때 : 95, 138
            1. Schedule의 static variable 인 scheduleNum보다 s2 (schedule 선택)이 크면 Exception을 발생시킴.
      2. ArrayFullException : line 173
         1. scheduleNum이 5 이상이면 exception을 발생시킴.
      3. InsufficientConditionException : line 252
         1. 반복적으로 값을 받아야 하므로, catch 문에서 재귀적으로 InsufficientLimit 메소드를 호출하도록 한다. activity가 ShowActivity라면 minAge 이하일 때 exception을 일으키고, ExtremeActivity이면 minHeight, min Weight 보다 작을 때 일으킨다.
   2. InputMismatchException : line 17, 45
7. **TravelScheduler**
   1. main
      1. FileInputStream을 통해서 Input을 받고, src폴더에 함께 있는 ActivityList, MemberList를 불러 array 형태로 불러낸다. Activity의 경우 nextLine으로 불러내서 “,”를 기준으로 split 해, [0]번째 index에 있는 Activity 종류 (activity, ExtremeActivity, ShowActivity) 에 맞게 activity를 생성해 activityList에 추가한다. MemberList 역시 “,”를 기준으로 split 해 Person object를 생성하고, personList에 추가한다.
      2. 메뉴 선택
         1. 메뉴에 없는 것을 선택할 때 exception을 일으킨다.
         2. Select schedule
            1. showSchedule method를 실행하고, 입력을 받는다. 입력이 Schedule.scheduleNum 보다 크면 Exception을 발생시킨다(빈 스케줄 이므로)
            2. Add Activity

addActivity method를 실행하고, tmp\_schedule에 넣는다. tmp\_schedule이 Null이면 처음으로 돌아가고, 아니면 scheduleList 에 추가한다.

* + - * 1. Remove Activity

day, time 을 입력받고 이 입력이 1-5 사이, 9-22 사이가 아니면 exception을 발생시킨다.

exception 없이 실행되면 schedule의 delActivity 메소드로 activity를 제거한다.

* + - * 1. Print schedule : printPlan method로 스케줄을 출력함.
      1. Edit schedule
         1. Make a new schedule

scheduleNum이 5 이상일 때 exception 발생

makeANewSchedule 메소드 실행해 scheduleList로 삼음.

* + - * 1. Copy an existing schedule

scheduleNum이 5 이상일 때 exception 발생

cophSchedule 메소드 실행.

* 1. custom exception
     1. WrongInput : 범위 안에 안들어온 값을 넣기 위한 exception
  2. other methods
     1. private static int limit(Scanner keyboard, int bound1, int bound2, String ask)
        1. 범위 안에 들어오지 않은 값에 대해 반복적으로 값을 받음. 범위 안에 안들어오면 exception을 발생시키고, catch 에서 재귀적으로 호출함.
     2. private static Person[] memberLimit(Scanner keyboard, Person[] personList)
        1. member 선택할 때 리스트 밖으로 넘어가는 값을 골랐을 경우 Exception을 발생시키고, 맞는 인풋을 넣을 때 까지 재귀적으로 호출함.
     3. private static int insufficientLimit(Schedule schedule, Activity[] activityList, Scanner keyboard)
        1. 선택한 activity의 조건이 member 들 간의 정보와 맞지 않을 때 조건에 맞는 activity를 고를 때 까지 반복함. 조건에 부합하지 않을 때 exception을 일으키고, catch에서 재귀적으로 insufficientLimit method를 호출.
     4. private static void showSchedule(Schedule[] scheduleList) : schedule List를 출력함.
     5. private static Schedule addActivity(Schedule schedule, Activity[] activityList, Scanner keyboard)
        1. 추가하고 싶은 날짜 시간을 받고, member들 간의 조건에 맞는지 확인한 후 맞는걸 고를 때 까지 반복함(insufficientLimit) checkPlan 메소드를 통해 그시간대에 activity가 존재하는지 확인하고, 있으면 InvalidAccessException 일으킴.
     6. private static Schedule[] makeANewSchedule(Schedule[] scheduleList, Person[] personList, Scanner keyboard)
        1. personList 출력 후 memberLimit 메소드를 통해 memberList를 만든다. 이름과 여행할 날을 입력받고 schedule을 추가함.
     7. private static Schedule[] copySchedule(Schedule[] scheduleList, Scanner keyboard)
        1. 복사할 Index를 받고 schedule 의 이름을 입력받아 copyconstructor 를 통해 하나 더 만든 후 scheduleList에 추가함.