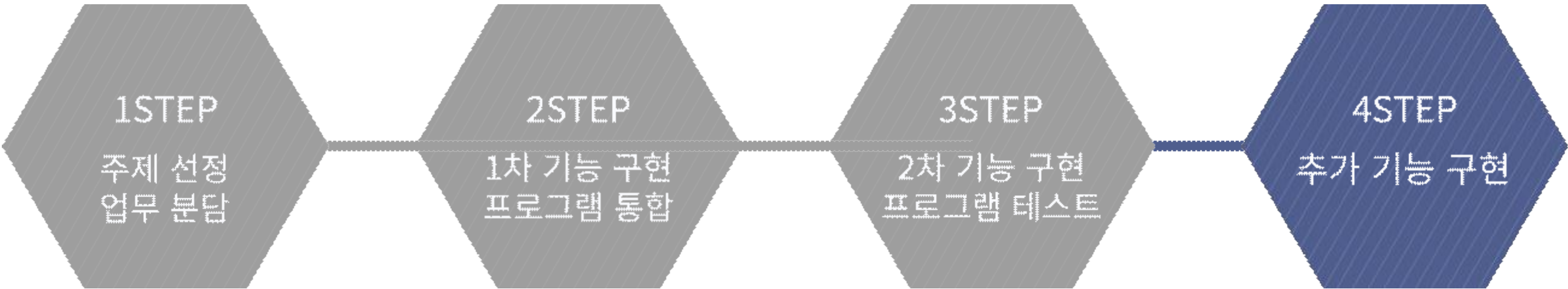


Accommodation reservation system with Java

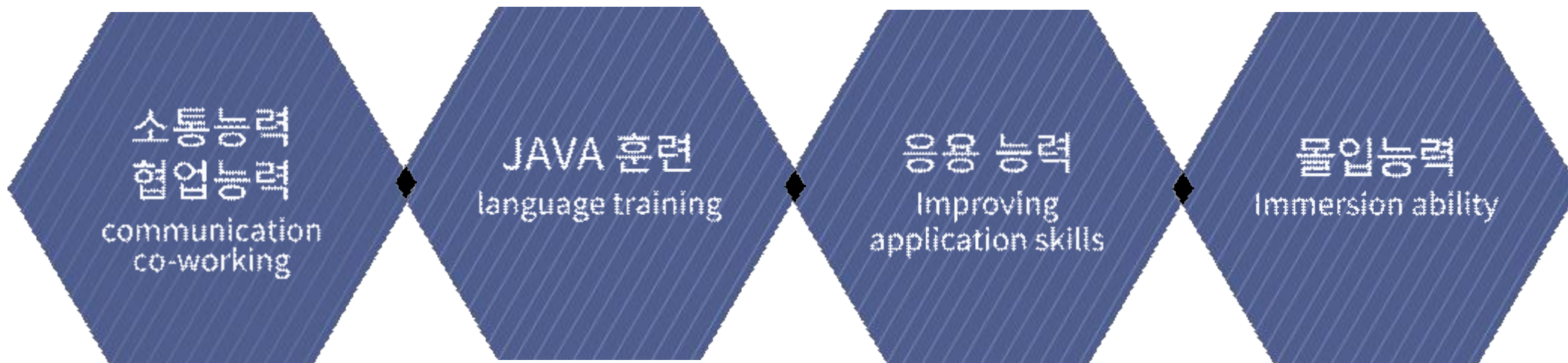
2023 Challenge | 에이콘 2조 | 김현수, 이윤정, 이정훈, 조은경

프로젝트 단계별 수행절차



2023/08/08 화		2023/08/09 수		2023/08/10 목	
오전	오후	오전	오후	오전	오후
주제 선정 및 업무 분담					
	1차 기능 구현	프로그램 통합		순서도 & 기능 도식화	
			2차 기능 구현		
				프로그램 테스트	
				추가 기능 구현	

Project Goal



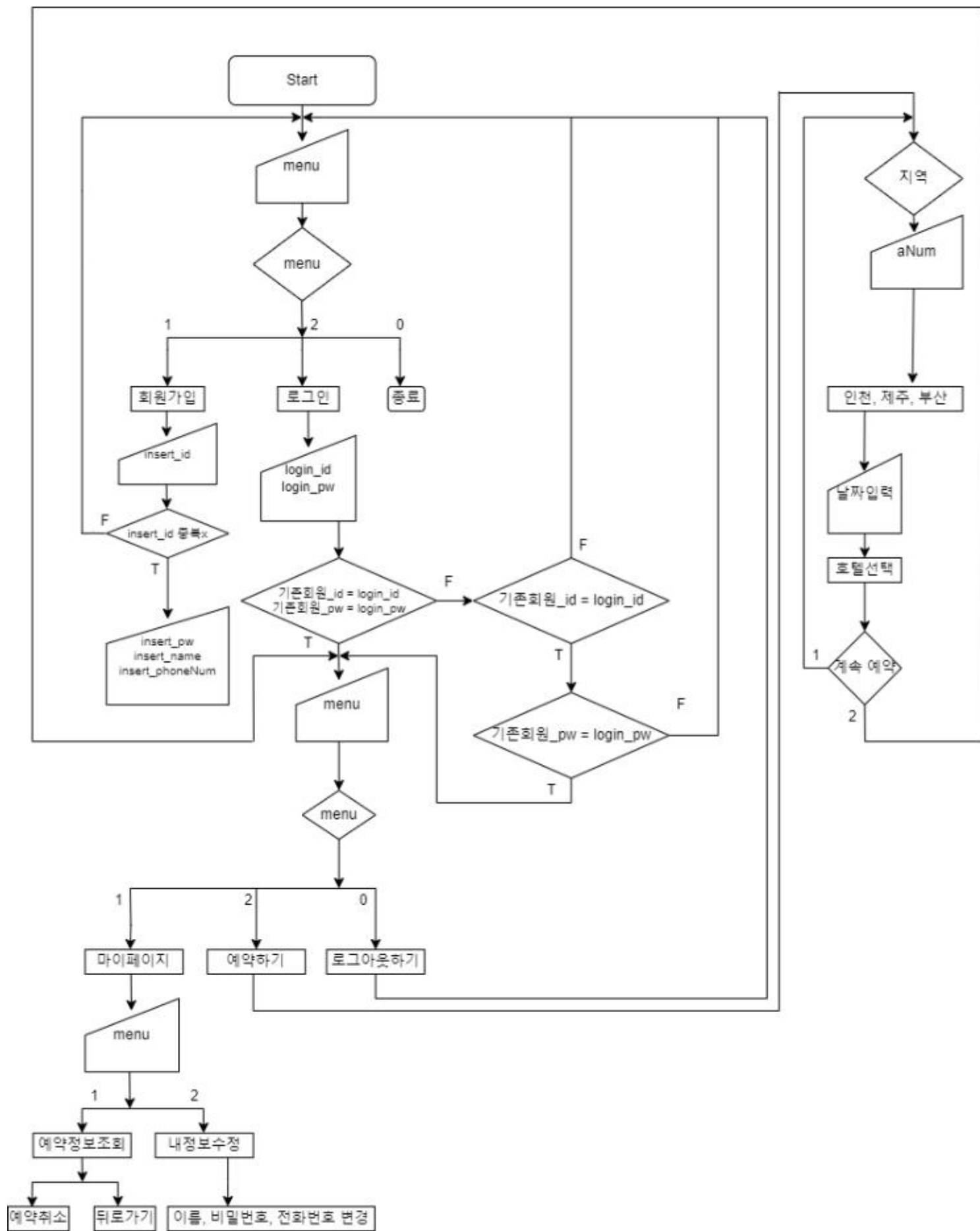
1. 소통을 통해 개인의 능력의 한계를 극복 (잘 듣기, 되묻기, 쌍방향 소통)
2. 팀 단위 프로젝트에서 필수불가결한 갈등 및 마찰 등을 조정하는 협업 능력 향상
3. 학습한 JAVA의 문법을 통해 직접 프로그램을 제작함으로써 JAVA 문법 숙련도 향상
4. 일상생활에서 접할 수 있는 내용을 프로그래밍 언어로 구현하는 응용능력 향상
5. 3일간의 짧은 기간 동안 하나의 프로젝트를 완성해나가는 과정에서 몰입 능력 향상

대표적인 숙소 예약 사이트

여기어때.

여기어때의 숙소 예약 시스템의 다양한 기능 중,
회원가입, 로그인, 개인정보 수정,
예약 정보 확인 등을 구현하는 것을 목표

| 프로그램 순서도



I 회원가입 & 로그인

```
public void memberInsert() {
    System.out.println("아이디 : ");
    insert_id = sc.nextLine();
    if (!isUniqueID(insert_id)) {
        System.out.println("이미 사용중인 아이디입니다.");
        return;
    }

    System.out.println("암호 : ");
    insert_pw = sc.nextLine();
    System.out.println("이름 : ");
    insert_name = sc.nextLine();
    System.out.println("전화번호 : ");
    insert_phoneNum = sc.nextLine();

    MemberInfo member = new MemberInfo(insert_id, insert_pw, insert_name, insert_phoneNum);
    memberInfos.add(member);
    System.out.println("가입 완료! \n");
}

private boolean isUniqueID(String id) {
    for (int i = 0; i < memberInfos.size(); i++) {
        if (memberInfos.get(i).getId().equals(id)) {
            return false;
        }
    }
    return true;
}
```

```
public String memberLogin() throws IOException, ClassNotFoundException {
    Scanner sc = new Scanner(System.in);
    System.out.println("아이디 : ");
    login_id = sc.nextLine();
    System.out.println("패스워드 : ");
    login_pw = sc.nextLine();
    String error = "fail";

    for (int i = 0; i < memberInfos.size(); i++) {
        MemberInfo member = memberInfos.get(i);
        if (member.getId().equals(login_id)) {
            if (member.getPw().equals(login_pw)) {
                System.out.println("로그인에 성공하셨습니다.");
                return login_id;
            } else {
                System.out.println("암호가 잘못되었습니다.");
                System.out.println("다시 로그인을 하세요");
                return error;
            }
        }
    }
    System.out.println("존재하지 않는 아이디입니다.");
    System.out.println("회원가입하세요.");
    return error;
}
```

```
case 2:
    login_id = handler.memberLogin();
    if (login_id != "fail") {
        menuSelect01();
    } else if (login_id == "fail") {
        System.out.println("회원가입하세요");
    }
    break;
```


| 예약 정보 조회 & 취소 | 내 정보 조회 & 변경

```
public static void myReservationSearch() {
    int cnt=1;
    if (reservation.size()==0) {
        System.out.println("예약된 정보가 없습니다.");
    }else {
        for(int i=0;i<reservation.size();i++) {
            Booking b = reservation.get(i);
            if(b.getId().equals(login_id)) {
                System.out.println((cnt)+". " + b);
                cnt++;
            }
        }
    }
}
```

```
public static void myReservationCancel() {
    System.out.println("삭제 예약정보번호를 입력하세요.");
    System.out.println("-----");
    myReservationSearch();
    System.out.println("-----");
    int num = sc.nextInt();
    int cnt = 1;
    for(int i=0;i<reservation.size();i++) {
        Booking b = reservation.get(i);
        if(b.getId().equals(login_id)) {
            if(num==cnt) {
                b.setId("null");
                b.setEmpty(false);
            }else {
                cnt++;
            }
        }
    }
}
```

```
public void ShowMyInfo() {
    for (int i = 0; i < memberInfos.size(); i++) {
        if (memberInfos.get(i).getId().equals(login_id) && memberInfos.get(i).getPw().equals(login_pw)) {
            login_id = memberInfos.get(i).getId();
            login_pw = memberInfos.get(i).getPw();
            login_name = memberInfos.get(i).getName();
            login_phoneNum = memberInfos.get(i).getPhonNum();
        }
    }
    System.out.println("이름 : " + login_name);
    System.out.println("아이디 : " + login_id);
    System.out.println("비밀번호 : " + login_pw);
    System.out.println("전화번호 : " + login_phoneNum);
}
```

```
public void ChangeMyPw() {
    System.out.println("변경할 비밀번호를 입력해주세요.");
    String newPw = sc.next();
    for (int i = 0; i < memberInfos.size(); i++) {
        if (memberInfos.get(i).getId().equals(login_id)) {
            login_pw = newPw;
            memberInfos.get(i).setPw(login_pw);
        } else if (login_pw == newPw) {
            System.out.println("비밀번호가 같습니다.");
        }
    }
}
```


| 숙소 리스트 생성

```
public class Booking implements Serializable {
    private String area, hotel, date, emp, id;
    private boolean empty;

    public Booking() {}
    public Booking(String area, String hotel, String date, boolean empty) {
        super();
        this.area = area;
        this.hotel = hotel;
        this.date = date;
        this.empty = empty;
    }
    public Booking(String id, String area, String hotel, String date, boolean empty) {
        super();
        this.id = id;
        this.area = area;
        this.hotel = hotel;
        this.date = date;
        this.empty = empty;
    }
    public String getId() {return id;}
    public String getArea() {return area;}
    public String getHotel() {return hotel;}
    public String getDate() {return date;}
    public boolean isEmpty() {return empty;}
    public void setId(String id) {this.id = id;}
    public void setEmpty(boolean empty) {this.empty = empty;}

    @Override
    public String toString() {
        if (empty == true) {emp = "예약";}
        else {emp = "예약가능";}
        return "ID:" + Main.login_id + " 지역:" + area + " 호텔:" + hotel + " 날짜:" + date + " 예약여부:" + emp;
    }
}
```

```
public class BookingList {

    static ArrayList<Booking> IncheonList = new ArrayList<>();
    static ArrayList<Booking> BusanList = new ArrayList<>();
    static ArrayList<Booking> JejuList = new ArrayList<>();

    static {
        // 인천
        for (int j = 0; j < BookingInfo.IncheonHotel.length; j++) {
            for (int k = 0; k < BookingInfo.date.length; k++) {
                Booking b = new Booking(BookingInfo.area[0], BookingInfo.IncheonHotel[j], BookingInfo.date[k], false);
                IncheonList.add(b);
            }
        }
        // 부산
        for (int j = 0; j < BookingInfo.BusanHotel.length; j++) {
            for (int k = 0; k < BookingInfo.date.length; k++) {
                Booking b = new Booking(BookingInfo.area[1], BookingInfo.BusanHotel[j], BookingInfo.date[k], false);
                BusanList.add(b);
            }
        }
        // 제주
        for (int j = 0; j < BookingInfo.JejuHotel.length; j++) {
            for (int k = 0; k < BookingInfo.date.length; k++) {
                Booking b = new Booking(BookingInfo.area[2], BookingInfo.JejuHotel[j], BookingInfo.date[k], false);
                JejuList.add(b);
            }
        }
    }
}
```

```
public class BookingInfo implements Serializable {

    static String[] area = { "인천", "부산", "제주" };
    static String[] date = new String[365];
    static String[] IncheonHotel = { "구월반원아시아드호텔", "네스트호텔", "인천공항디케이인천에어포트호텔", "구월호텔" };
    static String[] BusanHotel = { "해운대센트럴호텔", "엘모먼트공원", "신라스테이서부산", "해운대동백호텔" };
    static String[] JejuHotel = { "제주신라호텔", "글로스터호텔제주", "이든클리프호텔", "노블피시아호텔" };
    // String[] room;

    static {
        int cnt = 0;
        for (int i = 1; i <= 12; i++) {
            int k;
            if (i == 1 || i == 3 || i == 5 || i == 7 || i == 8 || i == 10 || i == 12) {
                k = 31;
            } else if (i == 2) {
                k = 28;
            } else {
                k = 30;
            }
            for (int j = 1; j <= k; j++) {
                date[cnt] = "2023/" + i + "/" + j;
                cnt++;
            }
        }
    }
}
```

| 숙소 예약, 조회

```
public static void reservation() {  
    // 예약된 정보 배열  
    loop: while (true) {  
        System.out.println("-----");  
        System.out.println("지역을 선택하세요");  
        for (int i = 0; i < BookingInfo.area.length; i++) {  
            System.out.print((i + 1) + "." + BookingInfo.area[i] + " ");  
        }  
        System.out.println();  
        aNum = sc.nextInt(); // 지역선택  
        area = BookingInfo.area[aNum - 1];  
  
        System.out.println("-----");  
        System.out.println("날짜를 입력하세요. ex) 0000/0/0");  
        date = sc.next(); // 날짜 선택  
  
        System.out.println("-----");  
        System.out.println("호텔을 선택하세요");  
        switch (aNum) {  
            case 1: // 인천  
                search(BookingInfo.IncheonHotel);  
                register(reservation);  
                break;  
            case 2: // 부산  
                search(BookingInfo.BusanHotel);  
                register(reservation);  
                break;  
            case 3: // 제주  
                search(BookingInfo.JejuHotel);  
                register(reservation);  
                break;  
            default:  
                System.out.println("잘못된 입력입니다");  
                break;  
        }  
  
        System.out.println("계속 예약하시겠습니까? 1.예 2.아니오");  
        aNum = sc.nextInt();  
        if (aNum != 1) {  
            break loop;  
        }  
    }  
}
```

```
// 지역 호텔 조회  
public static void search(String[] item) {  
    for (int i = 0; i < item.length; i++) {  
        System.out.print((i + 1) + "." + item[i] + " ");  
    }  
    System.out.println();  
    hNum = sc.nextInt(); // 호텔 선택  
    hotel = item[hNum - 1];  
}
```

```
// 예약 등록  
public static void register(ArrayList<Booking> item) {  
    for (int i = 0; i < item.size(); i++) {  
        Booking b = item.get(i);  
        if (b.getHotel().equals(hotel) && b.getDate().equals(date)) {  
            if (b.isEmpty() == false) {  
                b.setEmpty(true);  
                b.setId(login_id);  
                System.out.println("정상적으로 예약되었습니다.");  
            } else {  
                System.out.println("이미 예약된 방입니다.");  
            }  
        }  
    }  
}
```

```
// 등록된 예약 조회  
public static void registerSearch(ArrayList<Booking> item) {  
    for (int i = 0; i < item.size(); i++) {  
        Booking b = item.get(i);  
        if (b.isEmpty() == true) {  
            System.out.println(b);  
        }  
    }  
}
```


I 예약 정보 파일 저장 & 배열 불러오기

```
// 예약된 정보 파일에 저장
public static void FileInput(ArrayList<Booking> reservation) {
    try {
        BufferedWriter writer = new BufferedWriter(new FileWriter("res/List.txt"));
        // 문자열을 파일에 쓰기
        for(int i=0; i<reservation.size(); i++) {
            writer.write(reservation.get(i).toString()+"\n");
        }
        // 파일 쓰기 완료 후 닫기
        writer.close();
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```

FileInput : 숙소 전체 리스트를 파일에 저장

```
// 파일내용 다시 배열에 저장
public static void FileOutput(ArrayList<Booking> item) {
    try {
        BufferedReader bf = new BufferedReader(new FileReader("res/List.txt"));
        int cnt=0;
        while(true) {
            String data = bf.readLine();
            if( data == null) break;
            cnt++;
        }
        String[][] arr = new String[cnt][10];
        int index =0;
        bf = new BufferedReader(new FileReader("res/List.txt"));
        while(true) {
            String data = bf.readLine();
            if( data == null) break;
            arr[index] = data.split(" ");
            String b = "예약가능";
            if(arr[index][9].equals(b)){
                item.add( new Booking(arr[index][1], arr[index][3], arr[index][5], arr[index][7], false ));
            }else {
                item.add( new Booking(arr[index][1], arr[index][3], arr[index][5], arr[index][7], true ));
            }
            index++;
        }
    } catch (FileNotFoundException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```

FileOutput : 파일에 저장된 숙소 데이터를 다시 배열에 저장

I 영화 선택

```
void register() {
    System.out.println("---- 영화 정보 추가하기 ----");

    영화정보 m = new 영화정보("엘리멘탈", "피터 손", "109분");
    영화정보 m1 = new 영화정보("콘크리트 유토피아", "엄태화", "130분");
    영화정보 m2 = new 영화정보("오션하이머", "크리스토퍼 놀란", "180분");
    영화정보 m3 = new 영화정보("더 문", "김용화", "129분");
    영화정보 m4 = new 영화정보("명탐정코난-흑철의 여왕", "타치카와 유즈루", "110분");
    영화정보 m5 = new 영화정보("달까지근해-7510", "이한", "119분");

    list1.add(m);
    list1.add(m1);
    list1.add(m2);
    list1.add(m3);
    list1.add(m4);
    list1.add(m5);
}
```

```
void select() {
    System.out.println("---- 원하는 영화 선택하기 ----");
    Scanner sc = new Scanner(System.in);
    loop: while (true) {
        int n = sc.nextInt();
        switch (n) {
            case 1:
                System.out.println("엘리멘탈 영화를 선택하셨습니다");
                list1.get(0).updateCnt();
                break;
            case 2:
                System.out.println("콘크리트 유토피아 영화를 선택하셨습니다");
                list1.get(1).updateCnt();
                break;
            case 3:
                System.out.println("오션하이머 영화를 선택하셨습니다");
                list1.get(2).updateCnt();
                break;
            case 4:
                System.out.println("더 문 영화를 선택하셨습니다");
                list1.get(3).updateCnt();
                break;
            case 5:
                System.out.println("명탐정 코난-흑철의 여왕 영화를 선택하셨습니다");
                list1.get(4).updateCnt();
                break;
            case 6:
                System.out.println("달까지근해-7510 영화를 선택하셨습니다");
                list1.get(5).updateCnt();
                break;
            case 7:
                System.out.println("두표 종료");
                break loop;
        }
    }
}
```

```
void info() {
    System.out.println("---- 영화 정보 출력, 7번 누르면 종료 ----");
    for (int i = 0; i < list1.size(); i++) {
        System.out.println((i + 1) + " " + list1.get(i));
    }
}

void popular() {
    System.out.println("---- 인기순 정렬 ----");
    Collections.sort(list1, new Comparator<영화정보>() {
        @Override
        public int compare(영화정보 o1, 영화정보 o2) {
            // TODO Auto-generated method stub
            return o2.count - o1.count; // 내림차순 정렬, 오름차순 정렬하려면 o1과 o2 순서 바꾸기
        }
    });
    info();
}
```

```
public class 영화정보 {

    String name;
    String director;
    String running;
    int count;

    public 영화정보() {}

    public 영화정보(String name, String director, String running) {
        super();
        this.name = name;
        this.director = director;
        this.running = running;
    }

    public void updateCnt() {
        count++;
    }
}
```


감사합니다

2023 JAVA PROJECT

2023 Challenge | 에이콘