**동물 호텔 시스템**

2301110326 나예은

작성일: 2023. 10. 11 수

동물 호텔 시스템을 만들려고 한다. 시스템 요구사항은 아래와 같다.

1. 이름을 통해 체크인과 체크아웃을 한다.
2. 총 방의 개수는 2개이며, 한 방에 한 마리의 동물만이 들어올 수 있다.
3. 상속과 업캐스팅을 사용하여 시스템을 제작한다.
4. 동물의 종류는 개와 고양이 두 가지로 구분된다.
5. 호텔에서 각 동물이 체크아웃 할 때, 고양이는 ‘야옹~’을, 개는 ‘멍멍!’을 출력하며 나간다.
6. 객체지향 언어의 특성을 사용한다.

- 클래스는 총 5개(메인 클래스, 호텔기능, animal (부모), cat (자식), dog (자식))로 만들었다.

- 각 클래스의 설명

◉ MainStage 클래스

목적: MainStage 클래스는 호텔 관리 프로그램의 실행 흐름을 조절하고 사용자와의 상호 작용을 담당한다.

1. 무한 루프: while(true)를 함으로써 사용자가 프로그램을 종료하기 전까지 계속 무한 반복을 실행한다.
2. switch 문: switch(state)를 사용하여 state 값에 따라 다양한 작업을 수행한다. state 값에 따라 메뉴, 입실, 퇴실, 현황 메소드를 추가한다.
3. 메서드 호출: Hotel 클래스의 메서드를 호출하여 프로그램의 실행 흐름을 관리합니다. 각 case에서 해당 메서드를 호출하고, 호출 결과에 따라 stage 변수의 값을 변경합니다.

|  |
| --- |
| package animalH;  public class MainStage {  public static void main(String[] args) {  int stage = 0;    while(true) {  switch(stage) {  case 0:  stage = Hotel.Menu();  break;  case 1:  stage = Hotel.ChkIn();  break;  case 2:  stage = Hotel.ChkOut();  break;  case 3:  stage = Hotel.List();  break;  }  }  }  } |

◉ Hotel 클래스

- Hotel 클래스: Hotel 클래스는 호텔 관리 프로그램의 기능인 메뉴, 체크인, 체크아웃, 현황출력 메소드를 가지고 있다. 상호작용을 통해 해당하는 기능을 수행한다.

- Hotel 클래스 중 Menu() 메소드

목적: Menu() 메소드는 사용자에게 호텔 관리 프로그램의 메뉴를 출력하고, 사용자가 선택한 메뉴 번호를 반환하게 한다.

1. 메뉴 표시: 메뉴를 System.out.printf()를 사용하여 화면에 출력한다. 입실, 퇴실, 현황 출력 세가지 기능이 있으며, animal.chknum 값을 통해 현재 입실 및 퇴실한 동물의 총 수를 출력한다.
2. 사용자 입력 및 오류 처리: 사용자로부터 코드를 입력 받아 메뉴 번호를 읽는다. 입력값이 유효한 범위(1~3) 내에 있는지 확인하고, 그렇지 않을 시 오류 메세지를 출력하고 0을 반환하여 재시도를 하게 한다.
3. 메뉴 선택 반환: 유효한 값일 경우, 사용자가 선택한 번호 state를 반환한다.

|  |
| --- |
| static int Menu() {  System.out.printf(" [ 메뉴 ]\n 1. 입실\n 2. 퇴실\n 3. 현황\n + 입·퇴실 동물 수: %d\n", animal.chknum);  System.out.print(">> ");  int state = scan.nextInt();  if (state <= 0 || state > 3) {  System.out.println("<error> 입력 값 오류\n<system> 다시 입력해주세요.\n");  return 0;  }  System.out.println();  return state;  } |

- Hotel 클래스 중 ChkIn() 메소드

목적: ChkIn() 메소드는 호텔에 동물을 입실시키는 기능을 수행한다. 사용자가 선택한 동물의 종류와 이름을 입력하고, 해당 동물을 호텔에 입실시키는 것이다.

1. 사용자 인터페이스: 사용자에게 메뉴를 표시하고, 동물의 종류를 선택하고 이름을 입력하도록 안내합니다. 또한 잘못된 입력에 오류 처리를 수행하고, 메뉴로 돌아가도록 한다.
2. 호텔 상태 확인: am 배열을 통해 호텔의 현재 상태를 확인한다. 빈 방(am[i] == null)을 찾거나, 이미 같은 이름의 동물이 있을 경우 오류처리를 실행한다.   
   만실 상태 또한 확인한다.
3. 동물 객체 생성: 선택한 동물의 종류에 따라 cat 또는 dog객체를 생성하고, 호텔 배열에 저장한다.
4. 결과 출력: 입실에 성공할 경우, 성공 메세지와 함께 사용자에게 결과를 출력한다.
5. 오류 처리: 잘못된 입력이나 만실 상태 등에 대한 오류 메세지를 출력하고, 메뉴로 돌아가게 한다.

|  |
| --- |
| static int ChkIn() {  String n;  int chk = 999;  cat c;  dog d;    System.out.println(" [ 입실 ]\n \* 입실할 동물의 종류를 골라주세요.\n\n 1. cat\n 2. dog\n 3. 메뉴로 돌아가기");  System.out.print(">> ");  int k = scan.nextInt();  System.out.println();    if (k <= 0 || k > 3) {  System.out.println("<error> 입력 값\n<system> 다시 입력해주세요.\n");  return 1;  }  else if (k == 3) {  System.out.println("<system> 메뉴로 돌아갑니다.\n");  return 0;  }  else {  System.out.print(">> 동물의 이름을 입력해주세요: ");  n = scan.next();  for (int i = 0; i < am.length; i++) {  try {  if (am[i] == null) {  chk = i;  break;  }  else if (am[i].name.equals(n)) {  System.out.println("<error> 이름이 동일한 입실원이 존재\n<system> 메뉴로 돌아갑니다.\n");  return 0;  }  else if (am[i].chk == false && chk == 999) {  chk = i;  }  } catch (Exception e) {  }  }    if (chk == 999) {  System.out.println("<error> 만실로 인한 입실 불가\n<system> 메뉴로 돌아갑니다.\n");  return 0;  } else {  if (k == 1) {  c = new cat(n);  am[chk] = (animal)c;  } else {  d = new dog(n);  am[chk] = (animal)d;  }  System.out.printf("<success> %s이(가) 입실에 성공하였습니다!\n<system> 메뉴로 돌아갑니다.\n\n", n);  }  }  return 0;  } |

- Hotel 클래스 중 ChkOut() 메소드

목적: ChkOut() 메서드는 호텔에서 동물을 퇴실시키는 기능을 수행한다. 목적은 사용자가 동물의 이름을 입력하고, 해당 동물을 호텔에서 퇴실시키는 것이다.

1. 사용자 인터페이스: 코드는 사용자에게 동물의 이름을 입력하도록 안내하고, 입력을 받는다.
2. 호텔 상태 확인: am 배열을 통해 호텔의 현재 상태를 확인한다. 동물의 이름과 chk 상태를 비교하여, 퇴실 가능한 상태인지 확인한다.
3. 동물 객체와 클래스 확인: instanceof 연산자를 사용하여 동물이 cat인지, dog인지 확인하고, 각 동물 종류에 따라 적절한 행동을 수행한다.
4. 결과 출력: 퇴실에 성공한 경우, 성공 메시지와 함께 사용자에게 결과를 출력한다.
5. 오류 처리: 동물 이름이 호텔에 존재하지 않거나, 이름이 불일치하는 경우에 대한 오류 메시지를 출력하고, 사용자를 메뉴로 돌아가게 한다.

|  |
| --- |
| static int ChkOut() {  int key = 0;    System.out.print(" [ 퇴실 ]\n>> 퇴실할 동물의 이름을 적어주세요: ");  String n = scan.next();  System.out.println();    for (int i = 0; i < am.length; i++) {  if (am[i].name.equals(n) && am[i].chk == true) {  if (am[i] instanceof cat) {  am[i].chk = false;    System.out.printf("<success> %s이(가) 퇴실에 성공하였습니다!\n", am[i].name);  System.out.printf("\* %s: %s\n\n", am[i].name, cat.meow());    System.out.println("<system> 메뉴로 돌아갑니다.\n");  return 0;  }    else if (am[i] instanceof dog && am[i].chk == true) {  am[i].chk = false;    System.out.printf("<success> %s이(가) 퇴실에 성공하였습니다!\n", am[i].name);  System.out.printf("\* %s: %s\n\n", am[i].name, dog.wang());    System.out.println("<system> 메뉴로 돌아갑니다.\n");  return 0;  }  }  else {  key = 1;  }  }  if (key == 1) {  System.out.println("<error> 이름 불일치 혹은 존재하지 않음\n<system> 메뉴로 돌아갑니다. \n");  return 0;  }  return 0;  } |

- Hotel 클래스 중 List() 메소드

목적: List() 메서드는 현재 호텔의 입실 상태를 출력하는 기능을 수행한다. 목적은 호텔의 각 방에 대한 상태를 사용자에게 보여주는 것이다.

1. 사용자 인터페이스: 코드는 호텔 입실 상태를 화면에 출력하여 사용자에게 표시한다.
2. 호텔 상태 확인: am 배열을 통해 호텔의 각 방의 상태를 확인한다. 방이 비어있거나 (am[i] == null) 동물이 퇴실한 경우 (am[i].chk == false), 해당 방의 상태를 '비어있음'으로 표시한다. 그렇지 않은 경우 해당 방에 머물고 있는 동물의 이름을 출력한다.
3. 결과 출력: 호텔 입실 상태를 출력하여 사용자에게 보여준다.

|  |
| --- |
| static int List() {  System.out.println(" [ 현재 호텔 입실 현황 ]\n");  for (int i = 0; i < am.length; i++) {  if (am[i] == null || am[i].chk == false) {  System.out.printf("10%d호: 비어있음\n", i+1);  continue;  }  System.out.printf("10%d호: %s이(가) 사용중\n", i+1, am[i].name);  }  System.out.println();  return 0;  } |

- Hotel 클래스 전체 소스코드

|  |
| --- |
| package animalH;  import java.util.Scanner;  public class Hotel {  static Scanner scan = new Scanner(System.in);  static animal [] am = new animal[2];    static int Menu() {  System.out.printf(" [ 메뉴 ]\n 1. 입실\n 2. 퇴실\n 3. 현황\n + 입·퇴실 동물 수: %d\n", animal.chknum);  System.out.print(">> ");  int state = scan.nextInt();  if (state <= 0 || state > 3) {  System.out.println("<error> 입력 값 오류\n<system> 다시 입력해주세요.\n");  return 0;  }  System.out.println();  return state;  }    static int ChkIn() {  String n;  int chk = 999;  cat c;  dog d;    System.out.println(" [ 입실 ]\n \* 입실할 동물의 종류를 골라주세요.\n\n 1. cat\n 2. dog\n 3. 메뉴로 돌아가기");  System.out.print(">> ");  int k = scan.nextInt();  System.out.println();    if (k <= 0 || k > 3) {  System.out.println("<error> 입력 값\n<system> 다시 입력해주세요.\n");  return 1;  }  else if (k == 3) {  System.out.println("<system> 메뉴로 돌아갑니다.\n");  return 0;  }  else {  System.out.print(">> 동물의 이름을 입력해주세요: ");  n = scan.next();  for (int i = 0; i < am.length; i++) {  try {  if (am[i] == null) {  chk = i;  break;  }  else if (am[i].name.equals(n)) {  System.out.println("<error> 이름이 동일한 입실원이 존재\n<system> 메뉴로 돌아갑니다.\n");  return 0;  }  else if (am[i].chk == false && chk == 999) {  chk = i;  }  } catch (Exception e) {  }  }    if (chk == 999) {  System.out.println("<error> 만실로 인한 입실 불가\n<system> 메뉴로 돌아갑니다.\n");  return 0;  } else {  if (k == 1) {  c = new cat(n);  am[chk] = (animal)c;  } else {  d = new dog(n);  am[chk] = (animal)d;  }  System.out.printf("<success> %s이(가) 입실에 성공하였습니다!\n<system> 메뉴로 돌아갑니다.\n\n", n);  }  }  return 0;  }    static int ChkOut() {  int key = 0;    System.out.print(" [ 퇴실 ]\n>> 퇴실할 동물의 이름을 적어주세요: ");  String n = scan.next();  System.out.println();    for (int i = 0; i < am.length; i++) {  if (am[i].name.equals(n) && am[i].chk == true) {  if (am[i] instanceof cat) {  am[i].chk = false;    System.out.printf("<success> %s이(가) 퇴실에 성공하였습니다!\n", am[i].name);  System.out.printf("\* %s: %s\n\n", am[i].name, cat.meow());    System.out.println("<system> 메뉴로 돌아갑니다.\n");  return 0;  }    else if (am[i] instanceof dog && am[i].chk == true) {  am[i].chk = false;    System.out.printf("<success> %s이(가) 퇴실에 성공하였습니다!\n", am[i].name);  System.out.printf("\* %s: %s\n\n", am[i].name, dog.wang());    System.out.println("<system> 메뉴로 돌아갑니다.\n");  return 0;  }  }  else {  key = 1;  }  }  if (key == 1) {  System.out.println("<error> 이름 불일치 혹은 존재하지 않음\n<system> 메뉴로 돌아갑니다. \n");  return 0;  }      return 0;  }    static int List() {  System.out.println(" [ 현재 호텔 입실 현황 ]\n");  for (int i = 0; i < am.length; i++) {  if (am[i] == null || am[i].chk == false) {  System.out.printf("10%d호: 비어있음\n", i+1);  continue;  }  System.out.printf("10%d호: %s이(가) 사용중\n", i+1, am[i].name);  }  System.out.println();  return 0;  }  } |

◉ animal 클래스

목적: animal 클래스는 호텔의 입실한 동물을 나타내는 객체를 생성하기 위해 사용된다.

1. 속성: 클래스에는 세 가지 속성이 정의되어 있다.
2. name: 동물의 이름을 저장하는 문자열 속성이다.
3. chk: 동물의 입실 상태를 나타내는 부울 속성으로, true는 입실 상태를 나타낸다.
4. chknum: 정적으로 선언된 정수 속성으로, 현재 입실한 동물의 수를 추적한다.
5. 생성자: 클래스에는 animal 생성자가 정의되어 있다. 이 생성자는 동물의 이름을 인수로 받아 name 속성을 초기화하고, chk를 true로 설정한 후 chknum을 증가시킨다.

|  |
| --- |
| package animalH;  public class animal {  String name;  boolean chk;  static int chknum;    public animal (String name) {  this.name = name;    chk = true;  chknum += 1;  }  } |

◉ dog 클래스

목적: dog 클래스는 호텔 관리 프로그램에서 개를 나타내는 객체를 생성하기 위해 사용된다. 이 클래스는 animal 클래스를 상속받아 개 특유의 속성과 메서드를 추가한다.

1. 상속: dog 클래스는 animal 클래스를 상속받는다. 이것은 animal 클래스의 속성과 메서드를 dog 클래스에서 사용할 수 있게 해준다.
2. 추가 속성: dog 클래스에는 on이라는 부울 속성이 추가되었다. 이 속성은 개의 상태를 나타내며, true는 개가 활성 상태임을 나타낸다.
3. 생성자: dog 클래스에는 dog 생성자가 정의되어 있다. 이 생성자는 동물의 이름을 super(name)을 사용하여 부모 클래스의 생성자에 전달하고, on 속성을 true로 초기화한다.
4. 정적 메서드: wang()라는 정적 메서드가 정의되어 있다. 이 메서드는 "멍멍!" 문자열을 반환한다.

|  |
| --- |
| package animalH;  public class dog extends animal {  boolean on = true;    public dog(String name) {  super(name);  on = true;  }    public static String wang () {  return "멍멍!";  }  } |

◉ cat 클래스

목적: cat 클래스는 호텔 관리 프로그램에서 고양이를 나타내는 객체를 생성하기 위해 사용된다. 이 클래스는 animal 클래스를 상속받아 고양이 특유의 속성과 메서드를 추가한다.

1. 상속: cat 클래스는 animal 클래스를 상속받는다. 이것은 animal 클래스의 속성과 메서드를 cat 클래스에서 사용할 수 있게 해준다.
2. 추가 속성: cat 클래스에는 on이라는 부울 속성이 추가된다. 이 속성은 고양이의 상태를 나타내며, true는 고양이가 활성 상태임을 나타낸다.
3. 생성자: cat 클래스에는 cat 생성자가 정의되어 있다. 이 생성자는 동물의 이름을 super(name)을 사용하여 부모 클래스의 생성자에 전달하고, on 속성을 true로 초기화한다.
4. 정적 메서드: meow()라는 정적 메서드가 정의되어 있다. 이 메서드는 "야옹 ~" 문자열을 반환한다.

|  |
| --- |
| package animalH;  public class cat extends animal {  boolean on;    public cat(String name) {  super(name);  on = true;  }    public static String meow () {  return "야옹 ~";  }  } |

실행결과

- 시작 메뉴

텍스트, 폰트, 스크린샷, 번호이(가) 표시된 사진

자동 생성된 설명

- 첫번째 입실 성공 후 메뉴 출력 (입・퇴실 동물 수 카운트 증가)

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

- 두번째 입실 성공 후 메뉴 출력 (입・퇴실 동물 수 카운트 증가)

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

- 입실 → 메뉴로 돌아가기

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

- 입실 → 동일 이름일 시 오류 출력

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

- 입실 → 만실일 경우 오류 출력

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

- 퇴실 성공

텍스트, 스크린샷, 폰트, 디자인이(가) 표시된 사진

자동 생성된 설명

- 퇴실 → 이름 불일치 or 존재하지 않을 경우 오류 출력

텍스트, 폰트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

- 현황 출력 (기능 추가)

텍스트, 폰트, 스크린샷, 화이트이(가) 표시된 사진

자동 생성된 설명