**장난감 공장 실습문제 참고 문서**

ToyFactory\_Collection(선택사항).pdf 문서 상 애매하거나 작성되어있지 않은 내용 상세히 작성해둔 문서 입니다.

참고 바랍니다!

혹시 이해가 안가거나, 애매한 부분이 있다면 강사에게 질문!

**클래스 구성요소**

| **Toy Class**  필드  **private** String name; // 장난감 이름  **private** **int** age; // 사용 가능 연령  **private** **int** price; // 가격  **private** String color; // 색상  **private** String manufactureDate; // 제조일(문자열로 표현 - 20240222)  **private** Set<String> materials; // 재료 (중복되지 않는 값으로 저장)  기본생성자, 매개변수생성자(모든필드초기화용)  getter/setter  toString(), hashCode(), equals() 오버라이딩  **ToyFactory Class**  필드  **private Scanner sc = new Scanner(System.*in*);**  **private Set<Toy> toySet = new HashSet<Toy>();**  **// 중복된 Toy 객체가 없도록 저장할 컬렉션 중 Set**  **private Map<Integer, String> materialMap = new HashMap<Integer, String>();**  **// 재료가 저장되어있는 map**  기본생성자, **기본생성자에서 기본등록된 재료와, 현재 제작된 장난감 5개 작성 (아래 이미지 참고),**  displayMenu() 등 각종 메서드…    **Run Class**  main 메서드 ( ToyFactory 객체 생성 후 displayMenu() 호출 ) |
| --- |

| **\*\*\* 권장 사항 \*\*\***   1. **가능한 컬렉션의 주요 메서드를 사용해보기** 2. **예외처리 하기**   **\*\*\* 반드시 지켜야 할 내용 \*\*\***  1. 하나의 장난감에 같은 이름의 재료를 중복 저장할 수 없다.  -> 만약 중복으로 값이 들어온다면 덮어쓰기함  -> Toy라는 클래스 생성 시 재료 필드는 Set으로 작성  2. 똑같은 이름의 장난감을 만들 수 없다.  -> 처음에 장난감 이름을 먼저 받고 검사 후 같은 이름의 장난감이 없다면 추가 입력받도록 함  이미 존재한다면 이미 같은 이름을 가진 장난감이 존재한다는 메세지 띄우고 메서드 종료.  3. 장난감을 저장할 컬렉션은 중복된 장난감 객체가 있으면 안된다.  -> Set 이용(중복제거)  4. 재료는 { 고유번호 : 재료명 } 형식으로 작성되어 있어야한다  -> Map 이용{Key : Value}  5. 재료의 고유번호는 중복될 수 없다 (Map의 key는 중복 X)  **\*\*\* 기본 등록된 재료 / 현재 제작된 장난감 목록 \*\*\***  -> ToyFactory() 기본생성자에서 추가하기  -> Run의 main메서드에서 ToyFactory 객체 생성 시 기본생성자의 내용이 수행됨  -> Toy 객체 생성 시 재료 필드는 Set으로 저장.  -> 장난감 객체 생성 시 **재료 부분**은 **가변인자** 또는 **List.of()** 또는 **Arrays.asList()** 사용 할 수 있음   * **가변인자** : 메서드 호출 시 전달인자의 개수가 고정되어 있지 않고 가변적인 개수가 전달되어야 할 때 사용. (**자료형…변수명**)  | **메서드 정의**  public void printNumbers(**int... numbers**) {  for (int num : numbers) {  System.out.println(num);  }  }  **메서드 호출**  **printNumbers(1);**  // 1개의 인수 전달  **printNumbers(1, 2, 3);** // 3개의 인수 전달  **printNumbers();**  // 인수를 하나도 전달하지 않을 수도 있음 | | --- |      * **List.of()** : (Java 9이상부터 사용가능) 가변인자를 받아 전달받은 요소로 구성된 List 생성 후 반환  | List<Integer> list = **List.of(1, 4, 7)**;  System.out.println(list); // 출력: [1, 4, 7] | | --- |  * **Arrays.asList()** : (Java 8 이하 버전 사용 시)  | List<Integer> list = new ArrayList<Integer>(**Arrays.asList(1, 4, 7)**);  System.out.println(list); // 출력: [1, 4, 7] | | --- |   **\*\*\* 전체 장난감 목록은 아래 참고 이미지와 데이터 순서가 똑같지 않을 수 있습니다 \*\*\***    **\*\*\* 재료 추가/ 재료 제거에서 현재 등록된 재료 목록 출력할 땐 entrySet() 이용해보기 \*\*\***  Set<Map.Entry<K,V>> Map.entrySet() : 키와 값의 쌍으로 구성된 모든 Map.Entry 객체를 set에 담아서 리턴  ex)  for(Entry<Integer, String> entry : materialMap.entrySet()) {  System.out.println(entry.getKey() + " : " + entry.getValue());  }  getKey() : entry 객체에서 key만 얻어오기  getValue() : entry 객체에서 value만 얻어오기  **\*\*\* 재료 추가 시 재료저장용 컬렉션에 중복된 key 유무 체크 시 containsKey() 이용해보기 \*\*\***  boolean Map.containsKey(Object key) : 주어진 키가 있는지 확인하여 결과 리턴  **\*\*\* 재료명으로 재료 삭제 시 필요 메서드\*\*\***   1. 재료가 저장된 map을 순회하여 입력받은 재료명과 일치하는 재료가 있는지 확인해야하므로 entrySet() 이용 2. Entry객체에서 key와 value를 얻어오는 메서드 getValue() , getKey() 이용 3. 재료가 저장된 map의 데이터를 삭제해야하므로 remove() 이용   Object(Value) remove(Object key) : 주어진 키와 일치하는 Map.Entry 삭제, 삭제가 되면 삭제된 Value를 리턴 |
| --- | --- | --- | --- |
| **새로운 장난감 추가 메서드 변경사항**  **console 창에 아래와 같이 진행**  **앞서 재료 삭제로 { 3: 유리 } 를 제거를 진행한 상태라고 가정.**   * 새로운 장난감 추가 시 재료는 재료를 저장하고있는 컬렉션 이용 필수.  | <새로운 장난감 추가>  장난감 이름 : 빨간머리앤  사용 가능 연령 : 12  가격 : 15000  색상 : 빨간색  제조일 (YYYYMMDD 형식으로 입력) : 20240611  사용 가능한 재료 목록:  1 : 면직물  2 : 플라스틱  4 : 고무  추가할 재료의 번호를 입력하세요 (종료하려면 'q'를 입력하세요) : 3  잘못된 번호입니다. 다시 입력해주세요.  추가할 재료의 번호를 입력하세요 (종료하려면 'q'를 입력하세요) : 1  재료가 추가되었습니다 : 면직물  추가할 재료의 번호를 입력하세요 (종료하려면 'q'를 입력하세요) : 4  재료가 추가되었습니다 : 고무  추가할 재료의 번호를 입력하세요 (종료하려면 'q'를 입력하세요) : 4  이미 추가된 재료입니다. 다른 재료를 선택하세요.  추가할 재료의 번호를 입력하세요 (종료하려면 'q'를 입력하세요) : q  새로운 장난감이 추가되었습니다! | | --- |   **\*\*\* 재료 삭제 시 기존에 등록되어 있는 장난감의 재료가 삭제되지는 않습니다 \*\*\*** |