|  |  |  |
| --- | --- | --- |
| java2\_week14(보충)\_제네릭&컬렉션 | 학번 : 20237107 | 이름 : 하태영 |

* **개념 확인**

1. **[생성형 AI 활용]** 제시된 코드에서 배열 데이터 타입을 다양하게 적용할 수 있는 방법을 제시하고 테스트 하세요

class Cube {

**private int[] dim;**

public Cube(int[] dim){

this.dim=dim;

}

public void set(int index, int value ) {

this.dim[index] = value;

}

public int get(int index) {

return dim[index];

}

public String toString() {

return Arrays.*toString*(data);

}

}

|  |  |
| --- | --- |
| 생성형 AI | Perplexity |
| 질문 | 제시된 코드에서 배열 데이터 타입을 다양하게 적용할 수 있는 방법을 제시하고 테스트 하세요  class Cube {  **private int[] dim;**  public Cube(int[] dim){  this.dim=dim;  }  public void set(int index, int value ) {  this.dim[index] = value;  }  public int get(int index) {  return dim[index];  }  public String toString() {  return Arrays.*toString*(data);  }  } |
| 답변 | 배열 데이터 타입을 다양하게 적용하는 방법 제시된 코드에서는 int[]만 사용할 수 있어 배열 타입의 다양성이 부족합니다. 이를 해결하기 위해 \*\*제네릭(Generic)\*\*을 활용하면, 다양한 타입의 배열(int[], double[], String[] 등)을 지원할 수 있습니다.  **제네릭을 적용한 Cube 클래스 예시:**  java  **import** java.util.Arrays;  **class** Cube<T> {  **private** T[] dim;  **public** Cube(T[] dim) {  **this**.dim = dim;  }  **public** **void** set(**int** index, T value) {  **this**.dim[index] = value;  }  **public** T get(**int** index) {  **return** dim[index];  }  **public** String toString() {  **return** Arrays.toString(dim);  }  }  **테스트 코드 예시:**  java  **public** **class** Main {  **public** **static** **void** main(String[] args) {  *// Integer 배열*  Cube<Integer> intCube = **new** Cube<>(**new** Integer[3]);  intCube.set(0, 10);  intCube.set(1, 20);  intCube.set(2, 30);  System.out.println("Integer Cube: " + intCube);  *// Double 배열*  Cube<Double> doubleCube = **new** Cube<>(**new** Double[3]);  doubleCube.set(0, 1.1);  doubleCube.set(1, 2.2);  doubleCube.set(2, 3.3);  System.out.println("Double Cube: " + doubleCube);  *// String 배열*  Cube<String> stringCube = **new** Cube<>(**new** String[3]);  stringCube.set(0, "A");  stringCube.set(1, "B");  stringCube.set(2, "C");  System.out.println("String Cube: " + stringCube);  }  } 요약  * **제네릭(Generic) 클래스**로 구현하면 다양한 배열 데이터 타입을 적용할 수 있습니다. * 테스트를 통해 Integer, Double, String 등 여러 타입의 배열을 동일한 방식으로 사용할 수 있음을 확인할 수 있습니다. |

1. **[생성형 AI 활용]** 다음과 같은 코드가 컴파일 되지 않는다면 원인은 무엇일까요? 원인과 해결 방법을 함께 제시하세요

class MyAlgorithm {

public static <T> T max(T x, T y) {

return x > y ? x : y;

}

}

public class Test {

public static void main(String[] args) {

System.*out*.println(MyAlgorithm.*max*(34, 20));

}

}

|  |  |
| --- | --- |
| 생성형 AI | Perplexity |
| 질문 | 다음과 같은 코드가 컴파일 되지 않는다면 원인은 무엇일까요? 원인과 해결 방법을 함께 제시하세요  class MyAlgorithm {  public static <T> T max(T x, T y) {  return x > y ? x : y;  }  }  public class Test {  public static void main(String[] args) {  System.*out*.println(MyAlgorithm.*max*(34, 20));  }  } |
| 답변 | 컴파일 오류 원인 해당 코드는 다음과 같습니다:  java  **class** MyAlgorithm {  **public** **static** <T> T max(T x, T y) {  **return** x > y ? x : y;  }  }  **public** **class** Test {  **public** **static** **void** main(String[] args) {  System.out.println(MyAlgorithm.max(34, 20));  }  } 원인 설명  * **컴파일 오류의 원인**은 x > y 부분입니다. * 제네릭 타입 T는 Object의 하위 타입이기 때문에, > 연산자를 사용할 수 없습니다. 자바에서 >, < 등 비교 연산자는 원시 타입(primitive type)에만 사용할 수 있으며, 일반 객체(예: Integer, String 등)에는 사용할 수 없습니다[1](https://docs.oracle.com/javase/tutorial/java/generics/QandE/generics-answers.html)[2](https://stackoverflow.com/questions/26447035/compile-error-when-implement-a-simple-generic-function-in-java). * 따라서, x > y는 컴파일 오류를 발생시킵니다.  추가 설명  * 제네릭 타입 T에 대해 크기 비교를 하려면, T가 Comparable 인터페이스를 구현해야 하며, 비교는 compareTo() 메서드를 통해 해야 합니다[3](https://stackoverflow.com/questions/3347824/using-generics-to-create-max-function-that-returns-the-larger-one)[1](https://docs.oracle.com/javase/tutorial/java/generics/QandE/generics-answers.html)[2](https://stackoverflow.com/questions/26447035/compile-error-when-implement-a-simple-generic-function-in-java).  해결 방법 아래와 같이 T가 Comparable을 구현하도록 제네릭에 제한(bounded type parameter)을 추가하고, 비교 연산은 compareTo()로 수행해야 합니다.  java  **class** MyAlgorithm {  **public** **static** <T **extends** Comparable<T>> T max(T x, T y) {  **return** x.compareTo(y) > 0 ? x : y;  }  }  **public** **class** Test {  **public** **static** **void** main(String[] args) {  System.out.println(MyAlgorithm.max(34, 20)); *// 출력: 34*  }  }   * <T extends Comparable<T>>로 타입 제한을 추가하여, T 타입이 반드시 compareTo 메서드를 제공하도록 합니다. * x.compareTo(y) > 0을 통해 두 값을 비교합니다[3](https://stackoverflow.com/questions/3347824/using-generics-to-create-max-function-that-returns-the-larger-one)[1](https://docs.oracle.com/javase/tutorial/java/generics/QandE/generics-answers.html)[2](https://stackoverflow.com/questions/26447035/compile-error-when-implement-a-simple-generic-function-in-java).  요약  * **오류 원인**: 제네릭 타입 T에 대해 > 연산자를 사용할 수 없음. * **해결 방법**: T가 Comparable<T>를 구현하도록 제한하고, compareTo() 메서드로 비교한다. |

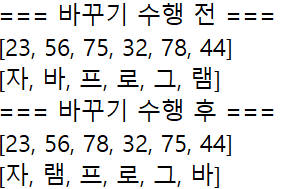
1. **[생성형 AI 활용]** List, Set, Map 인터페이스의 차이점을 제시하세요

|  |  |
| --- | --- |
| 생성형 AI | Perplexity |
| 질문 | List, Set, Map 인터페이스의 차이점을 제시하세요 |
| 답변 | List, Set, Map 인터페이스의 차이점  | **인터페이스** | **저장 구조 및 특징** | **순서 유지** | **중복 허용** | **주요 사용 예시** | | --- | --- | --- | --- | --- | | List | 순차적 저장, 인덱스로 접근 | O | O | 배열 대체, 순서 있는 데이터 관리 | | Set | 집합 구조, 값만 저장 | X | X | 중복 없는 데이터 집합, 유일값 관리 | | Map | Key-Value 쌍으로 저장 | Key: X | Key: X, Value: O | 키-값 쌍 데이터 관리, 빠른 검색 |  ****List****  * 데이터가 입력된 순서를 유지하며 저장됩니다. * 인덱스를 통해 각 요소에 접근할 수 있습니다. * 값의 중복을 허용합니다. * 대표 구현체: ArrayList, LinkedList 등[1](https://hahahoho5915.tistory.com/35)[2](https://popo015.tistory.com/102)[3](https://adjh54.tistory.com/138)[4](https://ccomccomhan.tistory.com/34)[5](https://jjungdev.tistory.com/11).  ****Set****  * 데이터의 저장 순서를 보장하지 않습니다(일부 구현체는 순서를 유지). * 값의 중복을 허용하지 않습니다. * 인덱스가 없어 직접 접근은 불가하며, Iterator 등을 통해 순회합니다. * 대표 구현체: HashSet, LinkedHashSet(순서 유지), TreeSet(정렬)[1](https://hahahoho5915.tistory.com/35)[2](https://popo015.tistory.com/102)[3](https://adjh54.tistory.com/138)[4](https://ccomccomhan.tistory.com/34)[5](https://jjungdev.tistory.com/11).  ****Map****  * Key와 Value의 쌍으로 데이터를 저장합니다. * Key는 중복을 허용하지 않으며, Value는 중복을 허용합니다. * 순서는 보장되지 않으나, LinkedHashMap 등은 입력 순서를 유지합니다. * 인덱스가 없으며, Key를 통해 Value에 접근합니다. * 대표 구현체: HashMap, LinkedHashMap, TreeMap[1](https://hahahoho5915.tistory.com/35)[2](https://popo015.tistory.com/102)[3](https://adjh54.tistory.com/138)[4](https://ccomccomhan.tistory.com/34)[5](https://jjungdev.tistory.com/11).  요약  * **List**: 순서 O, 중복 O, 인덱스 접근 O * **Set**: 순서 X(일부 구현체 O), 중복 X, 인덱스 접근 X * **Map**: Key-Value 구조, Key 중복 X, Value 중복 O, Key로 접근   각 인터페이스는 데이터의 저장 목적과 사용 방식에 따라 선택적으로 활용됩니다[1](https://hahahoho5915.tistory.com/35)[2](https://popo015.tistory.com/102)[3](https://adjh54.tistory.com/138)[4](https://ccomccomhan.tistory.com/34)[5](https://jjungdev.tistory.com/11). |

1. 아래의 프로그램 코드 ~~ 부분에 배열 원소 2개를 서로 교환하는 제네릭 메소드를 작성하고 테스트하세요.

class MyAlgorithm {

public static <T> void swap(T[] a, int i, int j) { //a[i]와 a[j]를 서로 교환



~~

}

}

public class Test {

public static void main(String[] args) {

Integer[] dim= {23, 56,75,32,78,44};

Character[] cdim = {'자', '바', '프','로','그','램'};

//이곳에 제시된 결과를 참고하여 나머지 부분을 채워 넣으세요

}

}

|  |
| --- |
| [프로그램 소스]  package practice4;  public class MyAlgorithm {  public static <T> void swap(T[] a, int i, int j) { // a[i]와 a[j]를 서로 교환  T temp = a[i];  a[i] = a[j];  a[j] = temp;  } }  package practice4;  import java.util.Arrays;  public class Test {  public static void main(String[] args) {  Integer[] dim= {23, 56,75,32,78,44};  Character[] cdim = {'자', '바', '프','로','그','램'};   System.*out*.println("=== 바꾸기 수행 전 ===");  System.*out*.println(Arrays.*toString*(dim));  System.*out*.println(Arrays.*toString*(cdim));   MyAlgorithm.*swap*(dim, 2, 4);  MyAlgorithm.*swap*(cdim, 1, 5);   System.*out*.println("=== 바꾸기 수행 후 ===");  System.*out*.println(Arrays.*toString*(dim));  System.*out*.println(Arrays.*toString*(cdim));   } } |
| [실행 결과] |

1. 타입 매개 변수 T를 가지는 클래스 MyMath를 작성합니다. MyMath에는 평균을 구하는 getAverage() 메소드를 추가합니다. 단, Byte, Short, Integer, Double, Float, Long 타입의 데이터에 대하여 평균을 구할 수 있도록 합니다.

public class Test{

public static void main(String[] args) {

Integer[] list = { 32,5,36,8,41,74,59,9,11};

Byte[] blist = {4,12,7,84,23,4};

MyMath<Integer> m = new MyMath<>();

MyMath<Byte> bm = new MyMath<>();

System.*out*.printf("average(Integer) : %.2f\n",m.getAverage(list));

System.*out*.printf("average(Byte) :%.2f\n", bm.getAverage(blist));

}

}

|  |
| --- |
| [프로그램 소스]  package practice5;  // Byte, Short, Integer, Double, Float, Long은 Number의 하위 클래스임으로, Number의 하위 클래스로 제한 class MyMath<T extends Number> {  public double getAverage(T[] numbers) {  // 데이터가 비어있는지 확인  if (numbers == null || numbers.length == 0) {  return 0.0;  }  // 총합 초기화  double sum = 0.0;   // 평균 계산  for (T num : numbers) {  sum += num.doubleValue(); // 실수형 타입으로 변환하여 총합에 더한다.  }  return sum / numbers.length;  } }  package practice5;  public class Test{  public static void main(String[] args) {  Integer[] list = { 32,5,36,8,41,74,59,9,11};  Byte[] blist = {4,12,7,84,23,4};   MyMath<Integer> m = new MyMath<>();  MyMath<Byte> bm = new MyMath<>();  System.*out*.printf("average(Integer) : %.2f\n",m.getAverage(list));  System.*out*.printf("average(Byte) :%.2f\n", bm.getAverage(blist));  } } |
| [실행 결과]  텍스트, 스크린샷, 폰트, 소프트웨어이(가) 표시된 사진  AI가 생성한 콘텐츠는 부정확할 수 있습니다. |

1. 다음과 같이 리스트가 생성되었을 때 제시된 문제를 해결하는 프로그램을 작성하세요

String[] s = { "사과", "배", "바나나" };

ArrayList<String> list = **new** ArrayList(Arrays.*asList*(s)); //배열을 리스트로 변환

|  |  |
| --- | --- |
|  | 1. “포도”를 리스트 마지막에 삽입 2. “자몽”을 인덱스 2에 삽입 3. 리스트에 저장된 데이터 출력 4. 인덱스 3에 저장된 데이터 출력 5. 인덱스 2에 저장된 데이터를 “메론”으로 변경 6. “배” 를 리스트 마지막에 삽입 7. “배”가 저장된 위치를 출력 8. 리스트에서 “바나나” 데이터 삭제 – remove(Object obj), remove(int index) 9. 리스트에 저장된 데이터 출력 |

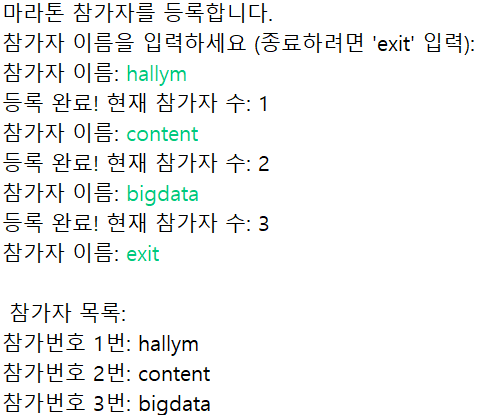
|  |
| --- |
| [프로그램 소스]  package practice6;  import java.util.ArrayList; import java.util.Arrays;  public class Main {  public static void write(ArrayList<String> list) {  System.*out*.println("------ 리스트 출력 ------");  list.forEach(System.*out*::println);  }   public static void main(String[] args) {  String[] s = { "사과", "배", "바나나" };  ArrayList<String> list = new ArrayList(Arrays.*asList*(s)); //배열을 리스트로 변환   // ① “포도”를 리스트 마지막에 삽입  list.add(list.size(), "포도");   // ② “자몽”을 인덱스 2에 삽입  list.add(2,"자몽");   // ③ 리스트에 저장된 데이터 출력  *write*(list);   // ④ 인덱스 3에 저장된 데이터 출력  String a = list.get(3);  System.*out*.println("----------------------");  System.*out*.println("인덱스 3 : " + a);   // ⑦ “배”가 저장된 위치를 출력  int idx = list.indexOf("배");  System.*out*.println("배가 저장된 위치 : " + idx);   // ⑤ 인덱스 2에 저장된 데이터를 “메론”으로 변경  list.set(2, "메론");   // ⑥ “배” 를 리스트 마지막에 삽입  list.add(list.size(), "배");   // ⑧ 리스트에서 “바나나” 데이터 삭제 – remove(Object obj), remove(int index)  list.remove("바나나");   // ⑨ 리스트에 저장된 데이터 출력  *write*(list);  } } |
| [실행 결과]  텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진  AI가 생성한 콘텐츠는 부정확할 수 있습니다. |

1. 두 자릿수 로또 번호 6개를 생성하는 프로그램을 작성하세요. 단, 중복을 허용하지 않으며 HashSet을 사용합니다



|  |
| --- |
| [프로그램 소스]  package practice7;  import java.util.HashSet; import java.util.Random; import java.util.Set;  public class Main {  public static void main(String[] args) {  Set<Integer> set = new HashSet<Integer>();  Random random = new Random();   while(set.size() < 6) {  int num = random.nextInt(90) + 10; // 10 ~ 99의 정수  set.add(num);  }  System.*out*.println("두 자수 로또 번호 6개를 출력합니다.");  System.*out*.println(set);  } } |
| [실행 결과]  텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진  AI가 생성한 콘텐츠는 부정확할 수 있습니다. |

1. Map을 사용하여 마라톤 참가자를 처리하는 프로그램을 작성하세요. 참가 번호는 선착순으로 사용하며 이름은 입력을 받습니다. key는 참가번호, value는 이름을 사용합니다. 참가 번호를 입력하면 이름을 출력합니다.



|  |
| --- |
| [프로그램 소스]  package practice8;  class Member {  private int number;  private String name;   public Member(int number, String name) {  this.number = number;  this.name = name;  }   @Override  public String toString() {  return "참가번호 " + number + "번 : " + name;  } }  package practice8;  import java.util.HashMap; import java.util.Map; import java.util.Scanner;  public class Main {  public static void main(String[] args){  Scanner sc = new Scanner(System.*in*);  Map<Integer, Member> map = new HashMap<>();  int count = 0;  System.*out*.println("마라톤 참가자를 등록합니다.");  System.*out*.println("참가자 이름을 입력해주세요 (종료하려면 'exit' 입력):");  while(true) {  System.*out*.print("참가자 이름 : ");  String name = sc.next();  if(name.equals("exit")) {  break;  }  count++;  map.put(count, new Member(count, name));  System.*out*.println("등록 완료! 현재 참가자 수 : " + count);  }  System.*out*.println("\n 참가자 목록:");  for (Member m : map.values()){  System.*out*.println(m);  }  } } |
| [실행 결과]  텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진  AI가 생성한 콘텐츠는 부정확할 수 있습니다. |

* **응용 프로그래밍**

1. 학생들의 정보를 ArrayList에 저장하고 검색할 수 있는 프로그램을 작성하세요. 학생들의 정보는 Student라는 클래스를 사용합니다. Student는 학생의 이름, 전화번호를 필드로 가지며 접근자와 설정자, 객체내용을 문자열로 반환하는 메소드로 구성합니다. ArrayListTest 클래스에서 학생들의 정보를 ArrayList에 추가, 검색, 삭제하는 간단한 메뉴를 제공합니다

public class ArrayListTest{

public static void main(String[] args) {

ArrayList<Student> list = new ArrayList<Student>();

Scanner sc = new Scanner(System.in);

for(int i=0;i<10; i++){

list.add(new Student("name"+i, "phone"+i));

}

for(Student s : list)

System.out.println(s);

//사용자에게 이름을 입력 받아 추가 – 본인작성

//사용자에게 이름을 입력 받아 전화번호 검색 – 본인 작성

//사용자에게 이름을 입력 받아 삭제 – 본인 작성

//모든 사용자 정보 출력 – 본인 작성

}

}

|  |
| --- |
| [프로그램 소스]  package practice9;  class Student {  private String name;  private String phone;   public Student(String name, String phone){  this.name = name;  this.phone = phone;  }   public String getName() {  return name;  }   public String getPhone() {  return phone;  }   @Override  public String toString() {  return "이름 : " + name + ", 전화번호 : " + phone;  } }  package practice9;  import java.util.ArrayList; import java.util.Scanner;  public class ArrayListTest{  public static void main(String[] args) {  ArrayList<Student> list = new ArrayList<Student>();  Scanner sc = new Scanner(System.*in*);   for (int i = 0; i < 10; i++) {  list.add(new Student("name" + i, "phone" + i));  }  for (Student s : list)  System.*out*.println(s);   boolean running = true;  while(running) {  System.*out*.println("1. 등록, 2. 검색, 3. 삭제, 4. 전체조회, 5. 종료");  int menu = 0;  System.*out*.print("메뉴 번호를 입력해주세요 : ");  menu = sc.nextInt();  switch (menu){  case 1: //사용자에게 이름을 입력 받아 추가 – 본인작성  System.*out*.print("이름을 입력하세요 : ");  String name = sc.next();  System.*out*.print("전화번호를 입력하세요 : ");  String phone = sc.next();  list.add(new Student(name, phone));  System.*out*.println("등록완료되었습니다.");  break;  case 2: //사용자에게 이름을 입력 받아 전화번호 검색 – 본인 작성  System.*out*.print("전화번호 검색할 이름을 입력해주세요 : ");  String search = sc.next();  boolean found = false;   for (Student s : list) {  if (s.getName().equals(search)) {  System.*out*.println(search + "의 전화번호 : " + s.getPhone());  found = true;  break;  }  }  if (!found) {  System.*out*.println(search + "의 정보를 찾을 수 없습니다.");  }  break;  case 3: //사용자에게 이름을 입력 받아 삭제 – 본인 작성  System.*out*.print("삭제할 이름을 입력해주세요 : ");  String del = sc.next();  Student toRemove = null;  // 향상된 for문(for-each)에서 직접 삭제하면 ConcurrentModificationException 발생할 수 있다.  for (Student s : list) {  if (s.getName().equals(del)) {  toRemove = s;  break;  }  }  if (toRemove != null) {  list.remove(toRemove);  System.*out*.println(del + "의 정보가 삭제되었습니다.");  } else {  System.*out*.println(del + "의 정보를 찾을 수 없습니다.");  }  break;  case 4: //모든 사용자 정보 출력 – 본인 작성  for (Student s : list) {  System.*out*.println(s);  }  break;  case 5:  running = false;  break;  default:  System.*out*.println("잘못된 입력하였습니다.");  }  }  } } |
| [실행 결과] |

1. 초등학생 영어 학습을 돕기 위한 프로그램을 HashMap을 이용하여 작성하시오.

- 영어 단어(key), 의미(value)로 구성한다

- 새로운 단어를 추가하거나 기존의 단어를 삭제할 수 있도록 한다

- HashMap에 저장된 모든 데이터 출력.

- HashMap에 다음의 데이터로 초기화한다

String[] word= {"actor", "age", "answer","arm", "autumn","air","airplane"};

String[] mean= {"배우", "나이","대답","팔","가을","공기","비행기"};

- 힌트) 삭제할 때 containsKey(), remove() 메소드 사용



|  |
| --- |
| [프로그램 소스]  package practice10;  class English {  private String word;  private String mean;   public English(String word, String mean){  this.word = word;  this.mean = mean;  }   @Override  public String toString() {  return mean;  } }  package practice10;  import java.util.\*;  public class Main {  public static void printMap(Map<String, English> map) {  System.*out*.println("====== Map에 저장된 결과 ======");  for (String key : map.keySet()) {  English value = map.get(key);  System.*out*.println("key=" + key + ", value=" + value);  }  }  public static void main(String[] args) {  Map<String, English> map = new HashMap<>();  Scanner sc = new Scanner(System.*in*);   String[] word = {"actor", "age", "answer", "arm", "autumn", "air", "airplane"};  String[] mean = {"배우", "나이", "대답", "팔", "가을", "공기", "비행기"};   for (int i = 0; i < word.length; i++) {  map.put(word[i], new English(word[i], mean[i]));  }   *printMap*(map);   System.*out*.print("\n삭제하고자 하는 영단어를 입력하시오 >> ");  String del = sc.next();   if (map.containsKey(del)) {  map.remove(del);  System.*out*.println(del + " 단어를 삭제합니다.");  } else {  System.*out*.println(del + " 단어가 존재하지 않습니다.");  }   *printMap*(map);   System.*out*.print("\n추가 하고자 하는 영단어를 입력하시오 >> ");  String addWord = sc.next();  System.*out*.print("단어의 뜻을 입력하시오 >> ");  String addMean = sc.next();  map.put(addWord, new English(addWord, addMean));   *printMap*(map);  } } |
| [실행 결과] |