**자료구조(박요한 교수님)**

**제네릭 클래스, 제네릭 메소드 만들기 과제**

컴퓨터공학과 유상현

**1. 프로그램 소스코드**

**App.java**

|  |
| --- |
| package Generic.App;  class Apple{  public String toString(){  return "사과";  }  }  class Orange{  public String toString(){  return "오렌지";  }  }  class Plastic{  public String toString(){  return "플라스틱";  }  }  class PrinterGeneric<T>{  private T ob;  public void set(T string){  ob = string;  }  public T get(){  return ob;  }  }  public class App {  public static <T> void PrintArray(T[] data){  int len = data.length;  for(int i = 0; i < len; i++){  System.out.print(data[i] + " ");  }  System.out.println();  }  public static void main(String[] args) {  PrinterGeneric<Apple> apple = new PrinterGeneric<>();  PrinterGeneric<Orange> orange = new PrinterGeneric<>();  PrinterGeneric<Plastic> plastic = new PrinterGeneric<>();  PrinterGeneric<String> object = new PrinterGeneric<>();  apple.set(new Apple());  orange.set(new Orange());  plastic.set(new Plastic());  object.set("파인애플");  System.out.println(apple.get());  System.out.println(orange.get());  System.out.println(plastic.get());  System.out.println(object.get());  // 2번 문제  Integer[] arrInt = {1,2,3,4,5};  Character[] arrChar = {'a','b','c','d','e'};  PrintArray(arrInt);  PrintArray(arrChar);  }  } |

**2. 실행 화면**

텍스트, 폰트, 스크린샷, 디자인이(가) 표시된 사진

자동 생성된 설명

**3. 느낀점(고찰)**

이번 제네릭 클래스와 메소드를 만들어 보면서 이를 활용하면 복잡했던 클래스들도 하나로 정리하여서 만들 수 있다는 것을 알게 되었다. 그리고 컴파일 시간에 에러를 잡을 수 있다는 것이 큰 매력으로 다가왔습니다.