```java
package Repository;

import DbUtils.DbProducts;
import Product.*;
import java.util.ArrayList;
import java.util.Collections;

import DbUtils.DbProducts;

public class ProductRepository {

    private ArrayList<Product> productList;
    private static ProductRepository productRepository=new ProductRepository();

    private ProductRepository() {
        this.productList= DbProducts.loadProductsData();

        System.out.println("załadowałem przez konstruktor ProductRepository");
    }

    public static ProductRepository getProductRepository() {
        return productRepository;
    }

    public static ArrayList<Product> getProductList() {
        return productRepository.productList;
    }

    public static void addProduct(String name, double price, String Id, String newCategory) {

            if (CategoryRepository.doesCategoryExists(newCategory)) {
                System.out.println("does= false dodaję produkt" );
                Product product=new Product(name, price,Id,newCategory);
                productRepository.productList.add(product);
                DbProducts.saveProducts(productRepository.productList);
                System.out.println("Product addet to list");
            } else {
                System.out.println("No such category, please add Category first" );
            }
    }

    public static void removeProduct(int indeks){
        if (indeks > 0 && indeks <= productRepository.productList.size()) {
            productRepository.productList.remove(indeks);
            DbProducts.saveProducts(productRepository.productList);
            } else {
            System.out.println("Indeks out of boundary, try again");
        }
    }

    public static void sortProductListBy(String sortMethod){
        switch(sortMethod){
            case "PRICE":
```

```
54                  Collections.sort(productRepository.productList, new
   CompareByPrice());
55                  break;
56              case "NAME":
57                  Collections.sort(productRepository.productList, new
   CompareByName());
58                  break;
59              case "CATEGORY":
60                  Collections.sort(productRepository.productList, new
   CompareByCategory());
61                  break;
62              case "DATE":
63                  Collections.sort(productRepository.productList, new
   CompareByDate());
64                  break;
65              default:
66                  System.out.println("Set sortMethod in User options");
67          }
68      }
69  }
70
```