Question 1: Smartphone Shop

a) The idea of my solution

Create two classes: Shop and Phone. In the class Shop we initialize format string to make printing as required in assignment. Also we create an array of phones to store objects of the class Phone. To add new phones we make method addPhones with the array of phones as an input parameter. If parameter in the class is equal to null, we just copy reference of the input array to the array in the class. If input array is equal to null, we just finish current method with "return" call and else we initialize new array with size which equal to sum of the sizes of the current array in the class and the input array, and then copy all the elements and copy a reference from the new array to current in the class. In printPhones we use format string from constants to print all the information about phones. In the Phone class we initialize all important attributes and generate for them getters and setters. We create constructor with string parameter for the name of the phone and inside generate an id using random generator. The id we calculate as 1000000000 plus random number from 0 to 8999999999 to get all the time 10-digit number. In the main class we just initialize 3 phones, initialize object of the class Shop, use addPhones method to add them initialized phones to shop and use printPhones method to print them on the screen.

b) Source code

- Shop.java

```
package uebung6.question1;

/**
 * Shop is a class, providing easy to use methods to add and print information
 * about smartphones out from the shop database.
 *
 * @author Andrii Dzhyrma
 */
public class Shop {

  // Output format string for the phones in array
  private static final String PHONE_OUTPUT_FORMAT_STRING = "Phone %d: %s [#%d]%n\t-CPU: %s%n\t-OS: %s%n\t-
    Resolution: %s%n\t-Storage: %s%n\t-Weight: %s%n";

  // The array to store the phones
  private Phone[] phones;

  /**
   * Inserts the specified elements to the end of the existing list of phones
   *
   * @param phones
   *            - elements to be inserted
   */
  public void addPhones(Phone[] phones) {
    // Check for existing of the array in the shop class
    if (this.phones == null)
      this.phones = phones;
    // Check for existing of the input array
    else if (phones == null)
      return;
    else {
      // Create new array with bigger size and copy all the elements to it
      Phone[] newPhones = new Phone[this.phones.length + phones.length];
      for (int i = 0; i < this.phones.length; i++)
        newPhones[i] = this.phones[i];
      for (int i = 0; i < phones.length; i++)
        newPhones[this.phones.length + i] = phones[i];
    }
  }

  /**
   * Prints out all the phones from the array
   */
```

```
  public void printPhones() {
    for (int i = 0; i < phones.length; i++)
      if (phones[i] != null)
        System.out.printf(PHONE_OUTPUT_FORMAT_STRING, i + 1,
            phones[i].getName(), phones[i].getId(),
            phones[i].getCpu(), phones[i].getOs(),
            phones[i].getResolution(), phones[i].getStorage(),
            phones[i].getWeight());
  }
}
```

- Phone.java

```
package uebung6.question1;

import java.util.Random;

/**
 * Phone is a class, providing easy to use methods to edit and read an important
 * parameters of the any smartphone.
 *
 * @author Andrii Dzhyrma
 */
public class Phone {

  // Initialize the empty string constant
  private static final String EMPTY_STRING = "";

  // Initialize the parameters of the phone
  private String cpu = EMPTY_STRING;
  private final long id;

  // Initialize the name and id variables as final
  private final String name;
  private String os = EMPTY_STRING;
  private String resolution = EMPTY_STRING;
  private String storage = EMPTY_STRING;
  private String weight = EMPTY_STRING;

  /**
   * Constructor of the class Phone.
   *
   * @param name
   *            - name of the phone.
   */
  public Phone(String name) {
    this.name = name;
    // Generate 10 digit random id
    id = (long) (new Random().nextDouble() * 8999999999d) + 1000000000;
  }

  /**
   * Determines the <tt>String</tt> value of the phone property <i>CPU</i>.
   *
   * @return the <tt>String</tt> value of the property.
   */
  public String getCpu() {
    return cpu;
  }

  /**
   * Determines the 10-digit <tt>long</tt> value of the phone property
   * <i>ID</i>.
   *
   * @return the <tt>long</tt> value of the property.
   */
  public long getId() {
    return id;
  }

  /**
   * Determines the <tt>String</tt> value of the phone property <i>Name</i>.
   *
   * @return the <tt>String</tt> value of the property.
   */
  public String getName() {
    return name;
  }
```

```java
/**
 * Determines the <tt>String</tt> value of the phone property <i>OS</i>.
 *
 * @return the <tt>String</tt> value of the property.
 */
public String getOs() {
  return os;
}

/**
 * Determines the <tt>String</tt> value of the phone property
 * <i>Resolution</i>.
 *
 * @return the <tt>String</tt> value of the property.
 */
public String getResolution() {
  return resolution;
}

/**
 * Determines the <tt>String</tt> value of the phone property
 * <i>Storage</i>.
 *
 * @return the <tt>String</tt> value of the property.
 */
public String getStorage() {
  return storage;
}

/**
 * Determines the <tt>String</tt> value of the phone property <i>Weight</i>.
 *
 * @return the <tt>String</tt> value of the property.
 */
public String getWeight() {
  return weight;
}

/**
 * Sets the phone property <i>CPU</i>.
 *
 * @param cpu
 *            - the new CPU value.
 */
public void setCpu(String cpu) {
  this.cpu = cpu;
}

/**
 * Sets the phone property <i>OS</i>.
 *
 * @param os
 *            - the new OS value.
 */
public void setOs(String os) {
  this.os = os;
}

/**
 * Sets the phone property <i>Resolution</i>.
 *
 * @param resolution
 *            - the new resolution value.
 */
public void setResolution(String resolution) {
  this.resolution = resolution;
}

/**
 * Sets the phone property <i>Storage</i>.
 *
 * @param storage
 *            - the new storage size value.
 */
public void setStorage(String storage) {
  this.storage = storage;
}
```

```
  /**
   * Sets the phone property <i>Weight</i>.
   *
   * @param weight
   *            - the new weight value.
   */
  public void setWeight(String weight) {
    this.weight = weight;
  }
}
```

- Main.java

```
package uebung6.question1;

public class Main {

  // Initialize the maximum value of phones
  static final int MAX_NUM_PHONES = 3;

  /**
   * @param args
   *             - no arguments will evaluate
   */
  public static void main(String[] args) {
    // Initialize the shop
    Shop shop = new Shop();

    // Initialize the array as container for numerous smrartphones.
    // Use a constant for the array length
    Phone[] phones = new Phone[MAX_NUM_PHONES];

    // Create a new phone and set its features and attributes
    Phone apple = new Phone("Apple iPhone 5");
    apple.setCpu("1.3 GHz dual core Apple A6");
    apple.setOs("iOS 6.0.1");
    apple.setResolution("640x1136");
    apple.setStorage("64 GB");
    apple.setWeight("112 g");

    Phone samsung = new Phone("Samsung Galaxy S III");
    samsung.setCpu("1.4 GHz quad-core Cortex-A9");
    samsung.setOs("Android 4.0 (Ice Cream Sandwich)");
    samsung.setResolution("720x1280");
    samsung.setStorage("32 GB");
    samsung.setWeight("133 g");

    Phone nokia = new Phone("Nokia Lumia 800");
    nokia.setCpu("1.4 GHz Scorpion");
    nokia.setOs("Microsoft Windows Phone 7.5 Mango");
    nokia.setResolution("480 x 800");
    nokia.setStorage("16 GB");
    nokia.setWeight("142 g");

    // Add the phone to the shop's phone array
    phones[0] = apple;
    phones[1] = samsung;
    phones[2] = nokia;

    // Add and print the phones to the screen
    shop.addPhones(phones);
    shop.printPhones();
  }
}
```

c) Test plan

Since no input were in the program, no test plan will be required.

d) The output of the program

```
Phone 1: Apple iPhone 5 [#2333939343]
       -CPU: 1.3 GHz dual core Apple A6
       -OS: iOS 6.0.1
       -Resolution: 640x1136
```

```
       -Storage: 64 GB
       -Weight: 112 g
Phone 2: Samsung Galaxy S III [#7331477472]
       -CPU: 1.4 GHz quad-core Cortex-A9
       -OS: Android 4.0 (Ice Cream Sandwich)
       -Resolution: 720x1280
       -Storage: 32 GB
       -Weight: 133 g
Phone 3: Nokia Lumia 800 [#8745423308]
       -CPU: 1.4 GHz Scorpion
       -OS: Microsoft Windows Phone 7.5 Mango
       -Resolution: 480 x 800
       -Storage: 16 GB
              -Weight: 142 g
```

Question 2: Interactive Electronic Shop

a) The idea of my solution.

We create three classes: Feature, Product and Shop. In the Feature class we create to string parameters: name and description. In the constructor we initialize those using input parameters. Also we create getters and setters for them, and toString method to convert object of the class to string using the initialized constant format string. In the Product class we create constant variables such as maximum number of features and string constants. To make adding and deleting features easier we initialize the integer value featureSize. In the constructor we use two input parameters: name and price. Also in the constructor we generate id as was described in previous solution. To add new features we create the addFeature method with an object of Feature type as an input parameter. In the method we check if number of features in the array is greater than max, and if not we adding new feature and increasing the featureSize value by one. Also we create the removeFeature method with an integer value as input parameter for an index of the feature to be removed. In the method we check if a feature with this index exists and if yes, remove and shift all features from the right by one to the left side and deacresing the featureSize value by one. Also we create the printFeatures method to optimize output choices when user should see all the features. For each an important parameter in the class we create getter and setter if required. In the Shop class we have also array, but with products instead of features. In the constructor we initialize only name of the shop. Adding, printing and removing methods are working with the same strategy as was described above with features. To get an reference for some special product we create getProduct method with the index of product as an input parameter. Also we create getter for the name parameter. Using switch statements and special printing methods we created user friendly dialogs as was written in the example from the assignment.

b) Source code

```java
/**
 * Feature is a class, providing easy to use methods for creating and editing
 * features with two properties "name" and "description".
 *
 * @author Andrii Dzhyrma
 */
public class Feature {

  // All the important constants
  private static final String TO_STRING_FORMAT_STRING = "[%s=%s]";

  // The name and the description properties
```

```
  private String name, description;

  /**
   * Constructor of the class Feature
   *
   * @param name
   *          - the name of the feature.
   * @param description
   *          - the description feature.
   */
  public Feature(String name, String description) {
    this.name = name;
    this.description = description;
  }

  /**
   * Determines the <tt>String</tt> value of the feature property
   * <i>Description</i>.
   *
   * @return the <tt>String</tt> value of the property.
   */
  public String getDescription() {
    return description;
  }

  /**
   * Determines the <tt>String</tt> value of the feature property <i>Name</i>.
   *
   * @return the <tt>String</tt> value of the property.
   */
  public String getName() {
    return name;
  }

  /**
   * Sets the feature property <i>Description</i>.
   *
   * @param description
   *          - the new description value.
   */
  public void setDescription(String description) {
    this.description = description;
  }

  /**
   * Sets the feature property <i>Name</i>.
   *
   * @param name
   *          - the new name value.
   */
  public void setName(String name) {
    this.name = name;
  }

  @Override
  public String toString() {
    return String.format(TO_STRING_FORMAT_STRING, name, description);
  }
}
```

- Product.java

```
package uebung6.question2;

import java.util.Random;

/**
 * Product is a class, providing easy to use methods for creating and editing
 * products with an any type of features.
 *
 * @author Andrii Dzhyrma
 */
public class Product {

  // All the important constants
  private static final int MAX_NUM_FEATURES = 10;
  private static final String TO_STRING_FORMAT_STRING = "Product %s (#%d) selected: [Price=%.2f]%s%n";
  private static final String DELIMITER_STRING = ", ";
```

```java
private static final String NO_FEATURES_STRING = "There is no feature in this product.";
private static final String FEATURE_LIST_ITEM_FORMAT_STRING = "[%d] %s%n";
// The identification number of the product
private final long id;

// The number of features in the product
private int featuresSize = 0;
// The array to store the features
private Feature[] features = new Feature[MAX_NUM_FEATURES];
// The name of the product
private String name;
// The price of the product
private double price;

/**
 * Constructor of the class Product.
 *
 * @param name
 *          - the name of the product.
 * @param price
 *          - the price of the product.
 */
public Product(String name, double price) {
  this.name = name;
  this.price = price;
  // Generate 10 digit random id
  id = (long) (new Random().nextDouble() * 8999999999d) + 1000000000;
}

/**
 * Inserts the specified element to the end of the existing list of features.
 *
 * @param feature
 *          - element to be inserted.
 * @return the result of insertion.
 */
public boolean addFeature(Feature feature) {
  if (featuresSize >= MAX_NUM_FEATURES)
    return false;
  features[featuresSize++] = feature;
  return true;
}

/**
 * Determines the <tt>Integer</tt> value of the amount of features in the
 * product.
 *
 * @return the <tt>Integer</tt> value of the property.
 */
public int getFeaturesSize() {
  return featuresSize;
}

/**
 * Determines the 10-digit <tt>long</tt> value of the product property
 * <i>ID</i>.
 *
 * @return the <tt>long</tt> value of the property.
 */
public long getId() {
  return id;
}

/**
 * Determines the <tt>String</tt> value of the product property <i>Name</i>.
 *
 * @return the <tt>String</tt> value of the property.
 */
public String getName() {
  return name;
}

/**
 * Determines the <tt>double</tt> value of the product property <i>Price</i>.
 *
 * @return the <tt>double</tt> value of the property.
 */
public double getPrice() {
```

```
    return price;
  }

  /**
   * Prints the list of all features to <tt>String</tt> value.
   *
   * @return the <tt>String</tt> which represents all features from the list.
   */
  public String printFeatures() {
    if (featuresSize == 0)
      return NO_FEATURES_STRING;
    String result = "";
    for (int i = 0; i < featuresSize; i++)
      result += String.format(FEATURE_LIST_ITEM_FORMAT_STRING, i, features[i]);
    return result;
  }

  /**
   * Removes feature from the given index in the list.
   *
   * @param index
   *          - the position of feature to be removed in the list.
   * @return the result of removal.
   */
  public boolean removeFeature(int index) {
    if (index < 0 || index >= featuresSize)
      return false;
    features[index] = null;
    for (int i = index + 1; i < featuresSize; features[i - 1] = features[i++])
      ;
    featuresSize--;
    return true;
  }

  /**
   * Sets the product property <i>Name</i>.
   *
   * @param name
   *          - the new name value.
   */
  public void setName(String name) {
    this.name = name;
  }

  /**
   * Sets the product property <i>Price</i>.
   *
   * @param price
   *          - the new price value
   */
  public void setPrice(double price) {
    this.price = price;
  }

  @Override
  public String toString() {
    String featuresString = "";
    for (int i = 0; i < featuresSize; i++)
      featuresString += DELIMITER_STRING + features[i].toString();
    return String.format(TO_STRING_FORMAT_STRING, name, id, price,
        featuresString);
  }
}
```

- Shop.java

```
package uebung6.question2;

/**
 * Shop is a class, providing easy to use methods such as addProduct,
 * removeProduct, getProduct and printProducts.
 *
 * @author Andrii Dzhyrma
 */
public class Shop {

  // All the important constants
  private static final int MAX_NUM_PRODUCTS = 100;
```

```java
private static final String NO_PRODUCTS_STRING = "There is no product in the shop.";
private static final String PRODUCT_LIST_ITEM_FORMAT_STRING = "[%d] %s (#%d)%n";
// The name of the shop
private final String name;

// The number of products in the shop
private int productsSize = 0;
// The array to store the products
private Product[] products = new Product[MAX_NUM_PRODUCTS];

/**
 * Constructor of the class Shop.
 *
 * @param name
 *          - the name of the shop.
 */
public Shop(String name) {
  this.name = name;
}

/**
 * Inserts new product to the shop's product list.
 *
 * @param product
 *          - product to be inserted.
 * @return the result of insertion.
 */
public boolean addProduct(Product product) {
  if (productsSize >= MAX_NUM_PRODUCTS)
    return false;
  products[productsSize++] = product;
  return true;
}

/**
 * Determines the <tt>String</tt> value of the shop's name.
 *
 * @return the <tt>String</tt> value of the name property.
 */
public String getName() {
  return name;
}

/**
 * Determines the <tt>Product</tt> value from the shop's list of products.
 *
 * @param index
 *          - the index of the product to be returned in the product list
 * @return the <tt>Product</tt> value of the chosen product.
 */
public Product getProduct(int index) {
  if (index < 0 || index >= MAX_NUM_PRODUCTS)
    return null;
  return products[index];
}

/**
 * Determines the <tt>Integer</tt> value of the amount of products in the
 * list.
 *
 * @return the amount of products in the list.
 */
public int getProductsSize() {
  return productsSize;
}

/**
 * Prints the list of all products to <tt>String</tt> value.
 *
 * @return the <tt>String</tt> which represents all products from the list.
 */
public String printProducts() {
  if (productsSize == 0)
    return NO_PRODUCTS_STRING;
  String result = "";
  for (int i = 0; i < productsSize; i++)
    result += String.format(PRODUCT_LIST_ITEM_FORMAT_STRING, i,
        products[i].getName(), products[i].getId());
```

```
    return result;
  }

  /**
   * Removes product from the given index in the list.
   *
   * @param index
   *          - the position of product to be removed in the list.
   * @return the result of removal.
   */
  public boolean removeProduct(int index) {
    if (index < 0 || index >= productsSize)
      return false;
    products[index] = null;
    // Shift all products after one removed to left
    for (int i = index + 1; i < productsSize; products[i - 1] = products[i++])
      ;
    productsSize--;
    return true;
  }
}
```

- Main.java

```
package uebung6.question2;

import io.Input;

/**
 * @author Andrii Dzhyrma
 */
public class Main {

  // The constant options to choose
  private static final String[] FEATURE_EDIT_OPTIONS = new String[] {
      "Delete an existing feature", "Add a new feature", "Return" };
  private static final String[] PRODUCT_EDIT_OPTIONS = new String[] {
      "Change product name", "Change product price",
      "Change product features", "Return" };
  private static final String[] SHOP_MAIN_OPTIONS = new String[] {
      "Print product list", "Add product", "Delete product",
      "Show product details", "Exit program" };

  // All the important string constants
  private static final String NAME_STRING = "name";
  private static final String PRICE_STRING = "price";
  private static final String SHOP_NAME_STRING = "My TestShop";
  private static final String SELECT_OPTION_STRING = "\nPlease select option:";
  private static final String PRODUCT_SELECTED_FEATURES_STRING = "The selected product has the following
    features:";
  private static final String ITEM_ADDED_ERROR_STRING = "The insertion of the new item failed! Probably
    you reached the maximum number...";
  private static final String ITEM_ADDED_STRING = "Item added successfully!";
  private static final String ITEM_REMOVED_STRING = "Item removed successfully!";
  private static final String ITEM_REMOVED_ERROR_STRING = "The removal of the item failed!";
  private static final String INVALID_INPUT_ERROR_STRING = "Invalid input. Please try again: ";

  private static final String PRODUCT_DELETE_FEATURE_REQUEST_STRING = "Please enter the index of the
    feature to be deleted: ";
  private static final String PRODUCT_FEATURE_REQUEST_STRING = "Please enter a feature (name=description)
    or hit enter to return: ";
  private static final String PRODUCT_NAME_REQUEST_STRING = "Please enter the product's name: ";
  private static final String PRODUCT_PRICE_REQUEST_STRING = "Please enter the product's price: ";
  private static final String SHOP_DELETE_PRODUCT_REQUEST_STRING = "Please select the index of the product
    to delete: ";
  private static final String SHOP_PRODUCT_REQUEST_STRING = "Please select the index of the product: ";

  private static final String SHOP_DESCRIPTION_FORMAT_STRING = "========================================%n
    %s%n========================================%n";
  private static final String OPTION_FORMAT_STRING = "[%d] %s%n";
  private static final String PRICE_FORMAT_STRING = "%.2f";
  private static final String PRODUCT_ATTRIBUTE_CHANGED_FORMAT_STRING = "Product %s changed
    successfully!%n";
  private static final String PRODUCT_CHANGE_ATTRIBUTE_FORMAT_STRING = "Current %s is: %s%nEnter new %s:";

  /**
   * @param args
   *            - no arguments will evaluate
```

```
 */
public static void main(String[] args) {
  Shop shop = new Shop(SHOP_NAME_STRING);
  Product pc = new Product("PC", 1199);
  pc.addFeature(new Feature("Brand", "Asus"));
  pc.addFeature(new Feature("USB", "4"));
  pc.addFeature(new Feature("Resolution", "1280x1024 pixels"));
  pc.addFeature(new Feature("Weight", "2.5 kg"));
  shop.addProduct(pc);
  Product tablet = new Product("Tablet", 549);
  tablet.addFeature(new Feature("Brand", "Asus"));
  tablet.addFeature(new Feature("USB", "1"));
  tablet.addFeature(new Feature("Resolution", "1280x800 pixels"));
  tablet.addFeature(new Feature("Weight", "635 g"));
  shop.addProduct(tablet);

  // Print out description of the current shop
  System.out.printf(SHOP_DESCRIPTION_FORMAT_STRING, shop.getName());
  // Make infinite cycle to ask user choose an option until he will choose
  // "Exit program"
  int choice;
  do {
    // Print out shop main options
    System.out.println(SELECT_OPTION_STRING);
    for (int i = 0; i < SHOP_MAIN_OPTIONS.length; i++)
      System.out.printf(OPTION_FORMAT_STRING, i + 1,
          SHOP_MAIN_OPTIONS[i]);
    // Read the user's choice
    choice = readInt(1, SHOP_MAIN_OPTIONS.length,
        INVALID_INPUT_ERROR_STRING);

    // Use switch statement to call correspondent methods
    switch (choice) {
    case 1: // Print product list
      System.out.println(shop.printProducts());
      break;
    case 2: // Add product
      printAddProductDialog(shop);
      break;
    case 3: // Delete product
      printDeleteProductDialog(shop);
      break;
    case 4: // Show product details
      printProductDetailsDialog(shop);
      break;
    case 5: // Exit program
      return;
    default:
      break;
    }
  } while (true);
}

// Prints dialog to add new feature into the chosen product
private static boolean printAddFeatureDialog(Product product,
    boolean isSuccessOutput) {
  // Initialize variables for the new feature creation
  String featureName;
  String featureDescription;
  Feature feature;

  // Print out request to write name and description for the new feature
  System.out.print(PRODUCT_FEATURE_REQUEST_STRING);
  // Read new feature represented as string
  String newFeatureString = Input.readString();
  // Initialize integer variable with the index of equal character
  if (newFeatureString.length() == 0)
    return false;
  // While the name or the description is empty, print out an error
  int equalIndex = newFeatureString.indexOf('=');
  while (equalIndex < 1 || equalIndex == newFeatureString.length() - 1) {
    System.out.println(INVALID_INPUT_ERROR_STRING);
    newFeatureString = Input.readString();
    equalIndex = newFeatureString.indexOf('=');
  }
  // Otherwise separate name and description values and make the new
  // feature
  featureName = newFeatureString.substring(0, equalIndex);
```

```
    featureDescription = newFeatureString.substring(equalIndex + 1);
    feature = new Feature(featureName, featureDescription);
    // Check for successful feature adding and print correspondent output
    if (!product.addFeature(feature))
      System.out.println(ITEM_ADDED_ERROR_STRING);
    else if (isSuccessOutput)
      System.out.println(ITEM_ADDED_STRING);
    return true;
  }

  // Prints dialog to add new product into the shop
  private static void printAddProductDialog(Shop shop) {
    // Initialize variables for the new product creation
    String name;
    Double price;
    Product product;

    // Print out request for the name of the new product
    System.out.print(PRODUCT_NAME_REQUEST_STRING);
    // Read the name
    name = Input.readString();
    while (name.length() == 0) {
      System.out.print(INVALID_INPUT_ERROR_STRING);
      name = Input.readString();
    }
    // Print out request for the price of the new product
    System.out.print(PRODUCT_PRICE_REQUEST_STRING);
    // Read the price
    price = Input.readDouble();
    while (price < 0) {
      System.out.print(INVALID_INPUT_ERROR_STRING);
      price = Input.readDouble();
    }
    // Convert price value to double value with only 2 digits after the dot
    price = Math.round(price * 100) / 100d;
    // Make the new product
    product = new Product(name, price);
    // Next loop will read and add new features to the new product until
    // user will input anything except enter
    while (printAddFeatureDialog(product, false))
      ;

    // Check for successful product adding and print correspondent output
    if (shop.addProduct(product))
      System.out.println(ITEM_ADDED_STRING);
    else
      System.out.println(ITEM_ADDED_ERROR_STRING);
  }

  // Prints dialog to delete a feature from the product
  private static void printDeleteFeatureDialog(Product product) {
    // Print out all features from the product
    System.out.println(product.printFeatures());
    // If there is no feature, make return
    if (product.getFeaturesSize() == 0)
      return;
    // Print out request for the index of the feature to be removed
    System.out.print(PRODUCT_DELETE_FEATURE_REQUEST_STRING);
    // Read the index
    int removeFeatureIndex = readInt(0, product.getFeaturesSize() - 1,
        INVALID_INPUT_ERROR_STRING);

    // Check for successful feature removal and print correspondent output
    if (product.removeFeature(removeFeatureIndex))
      System.out.println(ITEM_REMOVED_STRING);
    else
      System.out.println(ITEM_REMOVED_ERROR_STRING);
  }

  // Prints dialog to delete a product from the shop
  private static void printDeleteProductDialog(Shop shop) {
    // Print out all products from the shop
    System.out.println(shop.printProducts());
    // If there is no product, make return
    if (shop.getProductsSize() == 0)
      return;
    // Print out request for the index of the product to be removed
    System.out.print(SHOP_DELETE_PRODUCT_REQUEST_STRING);
```

```
    int removeIndex = readInt(0, shop.getProductsSize() - 1,
        INVALID_INPUT_ERROR_STRING);

    // Check for successful product removal and print correspondent output
    if (shop.removeProduct(removeIndex))
      System.out.println(ITEM_REMOVED_STRING);
    else
      System.out.println(ITEM_REMOVED_ERROR_STRING);
  }

  // Prints dialog to delete a feature, add, or go to the previous dialog
  private static void printProductChangeFeaturesDialog(Product product) {
    // Print out all features from the product
    System.out.println(PRODUCT_SELECTED_FEATURES_STRING);
    System.out.println(product.printFeatures());

    // Make loop to ask for the users choice until he/she will choose
    // "Return"
    int choice;
    do {
      // Print out feature options
      System.out.println(SELECT_OPTION_STRING);
      for (int i = 0; i < FEATURE_EDIT_OPTIONS.length; i++)
        System.out.printf(OPTION_FORMAT_STRING, i + 1,
            FEATURE_EDIT_OPTIONS[i]);
      // Read the user's choice
      choice = readInt(1, FEATURE_EDIT_OPTIONS.length,
          INVALID_INPUT_ERROR_STRING);

      // Use switch statement to call correspondent methods
      switch (choice) {
      case 1: // Delete an existing feature
        printDeleteFeatureDialog(product);
        break;
      case 2: // Add a new feature
        printAddFeatureDialog(product, true);
        break;
      default: // "Return" or any other options
        break;
      }
    } while (choice != FEATURE_EDIT_OPTIONS.length);
  }

  // Prints dialog to change the product name
  private static void printProductChangeNameDialog(Product product) {
    // Print out request for a new name of the product
    System.out.printf(PRODUCT_CHANGE_ATTRIBUTE_FORMAT_STRING, NAME_STRING,
        '"' + product.getName() + '"', NAME_STRING);
    // Read the new name until its size will be greater than zero
    String newName = Input.readString();
    while (newName.length() == 0) {
      System.out.print(INVALID_INPUT_ERROR_STRING);
      newName = Input.readString();
    }
    // Change the name of the product and print out confirmation message
    product.setName(newName);
    System.out.printf(PRODUCT_ATTRIBUTE_CHANGED_FORMAT_STRING, NAME_STRING);
  }

  // Prints dialog to change the product price
  private static void printProductChangePriceDialog(Product product) {
    // Print out request for a new price of the product
    System.out.printf(PRODUCT_CHANGE_ATTRIBUTE_FORMAT_STRING, PRICE_STRING,
        String.format(PRICE_FORMAT_STRING, product.getPrice()),
        PRICE_STRING);
    // Read the new price until it will be equal or greater than zero
    double newPrice = Input.readDouble();
    while (newPrice < 0) {
      System.out.print(INVALID_INPUT_ERROR_STRING);
      newPrice = Input.readDouble();
    }
    // Convert price value to double value with only 2 digits after the dot
    newPrice = Math.round(newPrice * 100) / 100d;
    // Change the price of the product and print out confirmation message
    product.setPrice(newPrice);
    System.out
        .printf(PRODUCT_ATTRIBUTE_CHANGED_FORMAT_STRING, PRICE_STRING);
  }
```

```
  // Prints dialog to show details of the product and gives possibility to
  // change them
  private static void printProductDetailsDialog(Shop shop) {
    // Print out all products from the shop
    System.out.println(shop.printProducts());
    // If there is no product, make return
    if (shop.getProductsSize() == 0)
      return;

    // Print out request for an index of product to be shown
    System.out.print(SHOP_PRODUCT_REQUEST_STRING);
    // Read the index of chosen product
    int index = readInt(0, shop.getProductsSize() - 1,
        INVALID_INPUT_ERROR_STRING);

    // Print out information about the product
    Product product = shop.getProduct(index);
    System.out.println(product);

    // Print out product editing options
    System.out.println(SELECT_OPTION_STRING);
    for (int i = 0; i < PRODUCT_EDIT_OPTIONS.length; i++)
      System.out.printf(OPTION_FORMAT_STRING, i + 1,
          PRODUCT_EDIT_OPTIONS[i]);
    // Read a user choice
    int choice = readInt(1, PRODUCT_EDIT_OPTIONS.length,
        INVALID_INPUT_ERROR_STRING);

    // Use switch statement to call correspondent methods
    switch (choice) {
    case 1: // Change product name
      printProductChangeNameDialog(product);
      break;
    case 2: // Change product price
      printProductChangePriceDialog(product);
      break;
    case 3: // Change product features
      printProductChangeFeaturesDialog(product);
      break;
    default: // "Return" or other options
      break;
    }
  }

  // Easy to use method for reading choice between min and max values. It will
  // raise an error if value is not in the input range.
  private static int readInt(int min, int max, String error) {
    int result = Input.readInt();
    while (result < min || result > max) {
      System.out.print(error);
      result = Input.readInt();
    }
    return result;
  }
}
```

c) Test plan

#	Aim	Input	Expected output
1	Common case	1, 5	**Normally printed list of products**
2	Common case	5	**Exit without an output**
3	Common case	3, 0, 1, 5	**Normally deleted product and printed new list of products**
4	Common case	2, Name, 100, F1=D1, <Enter>, 1, 5	**Normally added product and printed new list of products**
5	Common case	4, 0, 1, NewPC, 4, 0, 2, 1299, 4, 0, 3, 1, 1, 2, USB=5, 3, 4, 0, 4, 5	**Normally edited product**
6	Incorrect inputs	0, 6, 2, <Enter>, Name, -5, 10, =, =Desc1, Name1=, Name1=Desc1, <Enter>, 3, -1, 3, 2, 4, -1, 0, 0, 5, 1, <Enter>, NewPC, 4, 0, 2, -5, 10, 4, 0, 3, 0, 4, 1, -1, 4, 0, 2, =, =Desc1, Name1=, Name1=Desc1, 3, 5	**Print errors for an each incorrect input**
7	Extreme number of features	2, Name, 100, <Enter>, 4, 2, 3, 1, 2, Name1=Desc1, 3, 2, Name1, 100, N1=D1, N2=D2, N3=D3, N4=D4, N5=D5, N6=D6, N7=D7, N8=D8, N9=D9, N10=D10, N11=D11, 4, 3, 4, 5	**Normally print an information about number of features**
8	Extreme number of products	3, 0, 3, 0, 1, 3, 4, 5	**Normally print an information about number of products**

d) The output of the program

#1

```
========================================
  My TestShop
========================================

Please select option:
[1] Print product list
[2] Add product
[3] Delete product
[4] Show product details
[5] Exit program
1
[0] PC (#5153476164)
[1] Tablet (#9765615953)


Please select option:
[1] Print product list
```

#2

```
========================================
  My TestShop
========================================

Please select option:
[1] Print product list
[2] Add product
[3] Delete product
[4] Show product details
[5] Exit program
5
```

```
[2] Add product
[3] Delete product
[4] Show product details
[5] Exit program
5
```

#3

```
========================================
  My TestShop
========================================

Please select option:
[1] Print product list
[2] Add product
[3] Delete product
[4] Show product details
[5] Exit program
3
[0] PC (#8157258583)
[1] Tablet (#8764048327)

Please select the index of the product to
delete: 0
Item removed successfully!

Please select option:
[1] Print product list
[2] Add product
[3] Delete product
[4] Show product details
[5] Exit program
1
[0] Tablet (#8764048327)


Please select option:
[1] Print product list
[2] Add product
[3] Delete product
[4] Show product details
[5] Exit program
5
```

#4

```
========================================
  My TestShop
========================================

Please select option:
[1] Print product list
[2] Add product
[3] Delete product
[4] Show product details
[5] Exit program
2
Please enter the product's name: Name
Please enter the product's price: 100
Please enter a feature (name=description)
or hit enter to return: F1=D1
Please enter a feature (name=description)
or hit enter to return:
Item added successfully!

Please select option:
[1] Print product list
[2] Add product
[3] Delete product
[4] Show product details
[5] Exit program
1
[0] PC (#5153476164)
[1] Tablet (#9765615953)
[2] Name (#2468675445)


Please select option:
[1] Print product list
[2] Add product
[3] Delete product
[4] Show product details
[5] Exit program
5
```

#5

```
========================================
  My TestShop
========================================

Please select option:
[1] Print product list
[2] Add product
[3] Delete product
[4] Show product details
[5] Exit program
4
[0] PC (#2031207976)
[1] Tablet (#1463124533)

Please select the index of the product: 0
Product PC (#2031207976) selected:
```

#6

```
========================================
  My TestShop
========================================

Please select option:
[1] Print product list
[2] Add product
[3] Delete product
[4] Show product details
[5] Exit program
0
Invalid input. Please try again: 6
Invalid input. Please try again: 2
Please enter the product's name:
Invalid input. Please try again: Name
Please enter the product's price: -5
```

```
[Price=1199.00], [Brand=Asus], [USB=4],
[Resolution=1280x1024 pixels],
[Weight=2.5 kg]


Please select option:
[1] Change product name
[2] Change product price
[3] Change product features
[4] Return
1
Current name is: "PC"
Enter new name: NewPC
Product name changed successfully!

Please select option:
[1] Print product list
[2] Add product
[3] Delete product
[4] Show product details
[5] Exit program
4
[0] NewPC (#2031207976)
[1] Tablet (#1463124533)

Please select the index of the product: 0
Product NewPC (#2031207976) selected:
[Price=1199.00], [Brand=Asus], [USB=4],
[Resolution=1280x1024 pixels],
[Weight=2.5 kg]


Please select option:
[1] Change product name
[2] Change product price
[3] Change product features
[4] Return
2
Current price is: 1199.00
Enter new price: 1299
Product price changed successfully!

Please select option:
[1] Print product list
[2] Add product
[3] Delete product
[4] Show product details
[5] Exit program
4
[0] NewPC (#2031207976)
[1] Tablet (#1463124533)

Please select the index of the product: 0
Product NewPC (#2031207976) selected:
[Price=1299.00], [Brand=Asus], [USB=4],
[Resolution=1280x1024 pixels],
[Weight=2.5 kg]


Please select option:
[1] Change product name
[2] Change product price
```

```
Invalid input. Please try again: 10
Please enter a feature (name=description)
or hit enter to return: =
Invalid input. Please try again:
=Desc1
Invalid input. Please try again:
Name1=
Invalid input. Please try again:
Name1=Desc1
Please enter a feature (name=description)
or hit enter to return:
Item added successfully!

Please select option:
[1] Print product list
[2] Add product
[3] Delete product
[4] Show product details
[5] Exit program
3
[0] PC (#1917140642)
[1] Tablet (#2076227055)
[2] Name (#1738133199)

Please select the index of the product to
delete: -1
Invalid input. Please try again: 3
Invalid input. Please try again: 2
Item removed successfully!

Please select option:
[1] Print product list
[2] Add product
[3] Delete product
[4] Show product details
[5] Exit program
4
[0] PC (#1917140642)
[1] Tablet (#2076227055)

Please select the index of the product: -
1
Invalid input. Please try again: 0
Product PC (#1917140642) selected:
[Price=1199.00], [Brand=Asus], [USB=4],
[Resolution=1280x1024 pixels],
[Weight=2.5 kg]


Please select option:
[1] Change product name
[2] Change product price
[3] Change product features
[4] Return
0
Invalid input. Please try again: 5
Invalid input. Please try again: 1
Current name is: "PC"
Enter new name:
Invalid input. Please try again: NewPC
Product name changed successfully!
```

```
[3] Change product features
[4] Return
3
The selected product has the following
features:
[0] [Brand=Asus]
[1] [USB=4]
[2] [Resolution=1280x1024 pixels]
[3] [Weight=2.5 kg]


Please select option:
[1] Delete an existing feature
[2] Add a new feature
[3] Return
1
[0] [Brand=Asus]
[1] [USB=4]
[2] [Resolution=1280x1024 pixels]
[3] [Weight=2.5 kg]

Please enter the index of the feature to
be deleted: 1
Item removed successfully!

Please select option:
[1] Delete an existing feature
[2] Add a new feature
[3] Return
2
Please enter a feature (name=description)
or hit enter to return: USB=5
Item added successfully!

Please select option:
[1] Delete an existing feature
[2] Add a new feature
[3] Return
3

Please select option:
[1] Print product list
[2] Add product
[3] Delete product
[4] Show product details
[5] Exit program
4
[0] NewPC (#2031207976)
[1] Tablet (#1463124533)

Please select the index of the product: 0
Product NewPC (#2031207976) selected:
[Price=1299.00], [Brand=Asus],
[Resolution=1280x1024 pixels],
[Weight=2.5 kg], [USB=5]


Please select option:
[1] Change product name
[2] Change product price
[3] Change product features
[4] Return
```

```
Please select option:
[1] Print product list
[2] Add product
[3] Delete product
[4] Show product details
[5] Exit program
4
[0] NewPC (#1917140642)
[1] Tablet (#2076227055)

Please select the index of the product: 0
Product NewPC (#1917140642) selected:
[Price=1199.00], [Brand=Asus], [USB=4],
[Resolution=1280x1024 pixels],
[Weight=2.5 kg]


Please select option:
[1] Change product name
[2] Change product price
[3] Change product features
[4] Return
2
Current price is: 1199.00
Enter new price: -5
Invalid input. Please try again: 10
Product price changed successfully!

Please select option:
[1] Print product list
[2] Add product
[3] Delete product
[4] Show product details
[5] Exit program
4
[0] NewPC (#1917140642)
[1] Tablet (#2076227055)

Please select the index of the product: 0
Product NewPC (#1917140642) selected:
[Price=10.00], [Brand=Asus], [USB=4],
[Resolution=1280x1024 pixels],
[Weight=2.5 kg]


Please select option:
[1] Change product name
[2] Change product price
[3] Change product features
[4] Return
3
The selected product has the following
features:
[0] [Brand=Asus]
[1] [USB=4]
[2] [Resolution=1280x1024 pixels]
[3] [Weight=2.5 kg]


Please select option:
[1] Delete an existing feature
[2] Add a new feature
```

```
4

Please select option:
[1] Print product list
[2] Add product
[3] Delete product
[4] Show product details
[5] Exit program
5
```

```
[3] Return
0
Invalid input. Please try again: 4
Invalid input. Please try again: 1
[0] [Brand=Asus]
[1] [USB=4]
[2] [Resolution=1280x1024 pixels]
[3] [Weight=2.5 kg]

Please enter the index of the feature to
be deleted: -1
Invalid input. Please try again: 4
Invalid input. Please try again: 0
Item removed successfully!

Please select option:
[1] Delete an existing feature
[2] Add a new feature
[3] Return
2
Please enter a feature (name=description)
or hit enter to return: =
Invalid input. Please try again:
=Desc1
Invalid input. Please try again:
Name1=
Invalid input. Please try again:
Name1=Desc1
Item added successfully!

Please select option:
[1] Delete an existing feature
[2] Add a new feature
[3] Return
3

Please select option:
[1] Print product list
[2] Add product
[3] Delete product
[4] Show product details
[5] Exit program
5
```

#7

```
========================================
  My TestShop
========================================

Please select option:
[1] Print product list
[2] Add product
[3] Delete product
[4] Show product details
[5] Exit program
2
Please enter the product's name: Name
Please enter the product's price: 100
Please enter a feature (name=description)
or hit enter to return:
Item added successfully!

Please select option:
```

#8

```
========================================
  My TestShop
========================================

Please select option:
[1] Print product list
[2] Add product
[3] Delete product
[4] Show product details
[5] Exit program
3
[0] PC (#6142369130)
[1] Tablet (#1045963901)

Please select the index of the product to
delete: 0
Item removed successfully!
```

```
[1] Print product list
[2] Add product
[3] Delete product
[4] Show product details
[5] Exit program
4
[0] PC (#1536901173)
[1] Tablet (#1281661325)
[2] Name (#5118630896)

Please select the index of the product: 2
Product Name (#5118630896) selected:
[Price=100.00]


Please select option:
[1] Change product name
[2] Change product price
[3] Change product features
[4] Return
3
The selected product has the following
features:
There is no feature in this product.

Please select option:
[1] Delete an existing feature
[2] Add a new feature
[3] Return
1
There is no feature in this product.

Please select option:
[1] Delete an existing feature
[2] Add a new feature
[3] Return
2
Please enter a feature (name=description)
or hit enter to return: Name1=Desc1
Item added successfully!

Please select option:
[1] Delete an existing feature
[2] Add a new feature
[3] Return
3

Please select option:
[1] Print product list
[2] Add product
[3] Delete product
[4] Show product details
[5] Exit program
2
Please enter the product's name: Name1
Please enter the product's price: 100
Please enter a feature (name=description)
or hit enter to return: N1=D1
Please enter a feature (name=description)
or hit enter to return: N2=D2
Please enter a feature (name=description)
or hit enter to return: N3=D3
```

```
Please select option:
[1] Print product list
[2] Add product
[3] Delete product
[4] Show product details
[5] Exit program
3
[0] Tablet (#1045963901)

Please select the index of the product to
delete: 0
Item removed successfully!

Please select option:
[1] Print product list
[2] Add product
[3] Delete product
[4] Show product details
[5] Exit program
1
There is no product in the shop.

Please select option:
[1] Print product list
[2] Add product
[3] Delete product
[4] Show product details
[5] Exit program
3
There is no product in the shop.

Please select option:
[1] Print product list
[2] Add product
[3] Delete product
[4] Show product details
[5] Exit program
4
There is no product in the shop.

Please select option:
[1] Print product list
[2] Add product
[3] Delete product
[4] Show product details
[5] Exit program
5
```

```
Please enter a feature (name=description)
or hit enter to return: N4=D4
Please enter a feature (name=description)
or hit enter to return: N5=D5
Please enter a feature (name=description)
or hit enter to return: N6=D6
Please enter a feature (name=description)
or hit enter to return: N7=D7
Please enter a feature (name=description)
or hit enter to return: N8=D8
Please enter a feature (name=description)
or hit enter to return: N9=D9
Please enter a feature (name=description)
or hit enter to return: N10=D10
Please enter a feature (name=description)
or hit enter to return: N11=D11
The insertion of the new item failed!
Probably you reached the maximum
number...
Please enter a feature (name=description)
or hit enter to return:
Item added successfully!

Please select option:
[1] Print product list
[2] Add product
[3] Delete product
[4] Show product details
[5] Exit program
4
[0] PC (#1536901173)
[1] Tablet (#1281661325)
[2] Name (#5118630896)
[3] Name1 (#1672967692)

Please select the index of the product: 3
Product Name1 (#1672967692) selected:
[Price=100.00], [N1=D1], [N2=D2],
[N3=D3], [N4=D4], [N5=D5], [N6=D6],
[N7=D7], [N8=D8], [N9=D9], [N10=D10]


Please select option:
[1] Change product name
[2] Change product price
[3] Change product features
[4] Return
4

Please select option:
[1] Print product list
[2] Add product
[3] Delete product
[4] Show product details
[5] Exit program
5
```