5. Paveldėjimas

5.1. Darbo užduotis

U5_23. Juvelyrikos parduotuvė. Turite informaciją apie trijose juvelyrikos parduotuvėse esančius žiedus. Pirmoje eilutėje – pavadinimas, antroje – adresas, trečioje – telefonas. Parduotuvėje galima įsigyti žiedų, auskarų, grandinėlių. Sukurkite klasę „Juwelry“ (savybės - gamintojas, pavadinimas, metalas, svoris, praba, kaina), kurią paveldės „Ring” (savybė – dydis), „Earrings” (savybė – užsegimo tipas) ir „Collar” (savybė – ilgis).

• Raskite sunkiausią žiedą, auskarus ir grandinėlę. Ekrane atspausdinkite visą informaciją apie kiekvieną jų.

• Raskite, kiek aukščiausios prabos juvelyrinių gaminių yra kiekvienoje parduotuvėje, rezultatą atspausdinkite ekrane. Informacija apie lietuviškas prabas: platinos – 950; aukso – 375, 585 ir 750; sidabro – 800, 830 ir 925; paladžio – 500 ir 850.

• Ar yra tokių juvelyrinių dirbinių, kurių galima įsigyti visose juvelyrinėse parduotuvėse? Atspausdinkite visą informaciją apie juos faile „Visur.csv“.

• Sudarykite juvelyrinių dirbinių, pigesnių nei 300 eurų, sąrašą, išrikiuokite pagal gamintoją, pavadinimą ir kainą. Visus duomenis apie juos įrašykite į failą „300.csv“.

5.2. Programos tekstas

Quality.cs

```
namespace U5_23
{
    /// <summary>
    /// This enum is used to determine the quality of the Metal
    /// </summary>
    public enum Quality
    {
        Platinum = 950, Gold = 750, Silver = 925, Palladium = 850
    }
}
```

Rings.cs

```
using System;

namespace U5_23
{
    class Ring : Juwelry
    {
        public int Size { get; set; }
        /// <summary>
        /// Constructor for ring
        /// </summary>
        /// <param name="Size"> ring size </param>
        /// <param name="manufacturer"> manufacturer </param>
        /// <param name="name"> name </param>
        /// <param name="metal"> metal </param>
        /// <param name="weight"> weight </param>
        /// <param name="fineness"> fineness </param>
        /// <param name="price"> price </param>
        public Ring(int Size, string manufacturer, string name, string metal,
double weight, int fineness, double price) : base(manufacturer, name, metal,
weight, fineness, price)
        {
            this.Size = Size;
        }
```

```
        /// <summary>
        /// Method used to override ToString()
        /// </summary>
        /// <returns> returns formated string line </returns>
        public override string ToString()
        {
            return string.Format("| {0,-16} | {1,-15} | {2,-10} | {3,-6} | {4,-8}
| {5,-6} | {6,-4} | {7,-4} | {7,-6} |", Manufacturer, Name, Metal, Weight,
Fineness, Price, Size, String.Empty);
        }
    }
}
```

Earrings.cs

```
namespace U5_23
{
    class Earrings : Juwelry
    {
        public string EarringBack { get; set; }
        /// <summary>
        /// Constructor for earrings
        /// </summary>
        /// <param name="EarringBack"> Earrings locking type </param>
        /// <param name="manufacturer"> manufacturer </param>
        /// <param name="name"> name </param>
        /// <param name="metal"> metal </param>
        /// <param name="weight"> weight </param>
        /// <param name="fineness"> fineness </param>
        /// <param name="price"> price </param>
        public Earrings(string EarringBack, string manufacturer, string name,
string metal, double weight, int fineness, double price) : base(manufacturer,
name, metal, weight, fineness, price)
        {
            this.EarringBack = EarringBack;
        }
        /// <summary>
        /// Method used to override ToString()
        /// </summary>
        /// <returns> returns formated string line </returns>
        public override string ToString()
        {
            return string.Format("| {0,-16} | {1,-15} | {2,-10} | {3,-6} | {4,-8}
| {5,-6} | {6,-4} | {7,-4} | {6,-6} |", Manufacturer, Name, Metal, Weight,
Fineness, Price, " ", EarringBack);
        }
    }
}
```

Collar.cs

```
namespace U5_23
{
    class Collar : Juwelry
    {
        public double Length { get; set; }
        /// <summary>
        /// Constructor for collars
        /// </summary>
        /// <param name="Length"> Collar length </param>
        /// <param name="manufacturer"> manufacturer </param>
        /// <param name="name"> name </param>
        /// <param name="metal"> metal </param>
        /// <param name="weight"> weight </param>
        /// <param name="fineness"> fineness </param>
```

```
        /// <param name="price"> price </param>
        public Collar(double Length, string manufacturer, string name, string
metal, double weight, int fineness, double price) : base(manufacturer, name,
metal, weight, fineness, price)
        {
            this.Length = Length;
        }
        /// <summary>
        /// Method used to override ToString()
        /// </summary>
        /// <returns> returns formated string line </returns>
        public override string ToString()
        {
            return string.Format("| {0,-16} | {1,-15} | {2,-10} | {3,-6} | {4,-8}
| {5,-6} | {6,-4} | {6,-4} | {7,-6} |", Manufacturer, Name, Metal, Weight,
Fineness, Price," ", Length);
        }
    }
}
```

Juwelry.cs

```
namespace U5_23
{
    /// <summary>
    /// Class for Jewelry
    /// </summary>
    public class Juwelry
    {
        public string Manufacturer { get; set; }
        public string Name { get; set; }
        public string Metal { get; set; }
        public double Weight { get; set; }
        public int Fineness { get; set; }
        public double Price { get; set; }
        /// <summary>
        /// Constructor for Jewelry
        /// </summary>
        /// <param name="manufacturer"></param>
        /// <param name="name"></param>
        /// <param name="metal"></param>
        /// <param name="weight"></param>
        /// <param name="fineness"></param>
        /// <param name="price"></param>
        public Juwelry(string manufacturer, string name, string metal, double
weight, int fineness, double price)
        {
            Manufacturer = manufacturer;
            Name = name;
            Metal = metal;
            Weight = weight;
            Fineness = fineness;
            Price = price;
        }
        /// <summary>
        /// Method used to find weight
        /// </summary>
        /// <returns> weight</returns>
        public double GetWeight()
        {
            return Weight;
        }
        /// <summary>
        /// Method used to override ToString()
        /// </summary>
```

```
        /// <returns> returns formated string line </returns>
        public static string PrintFormat()
        {
            return string.Format("| {0, -16} | {1,-15} | {2,-10} | {3, -6} | {4, -
8} | {5, -6} | {6, -4} | {7, -4} | {8, -6} |", "Manufacturer", "Name", "Metal",
"Weight", "Fineness", "Price", "Size", "Lock", "Length");
        }
        /// <summary>
        /// Comparing two elements
        /// </summary>
        /// <param name="lhs"> left element </param>
        /// <param name="rhs"> right element </param>
        /// <returns> true if their Name is the same </returns>
        public static bool operator == (Juwelry lhs, Juwelry rhs)
        {
            return lhs.Name == rhs.Name;
        }
        /// <summary>
        /// Comparing two elements
        /// </summary>
        /// <param name="lhs"> left element </param>
        /// <param name="rhs"> right element </param>
        /// <returns> true their Name is not the same </returns>
        public static bool operator != (Juwelry lhs, Juwelry rhs)
        {
            return lhs.Name != rhs.Name;
        }
    }
}
```

JuwelryContainer.cs

```
using System;

namespace U5_23
{
    /// <summary>
    /// Shop container class
    /// </summary>
    class JuwelryContainer
    {
        //Ring list
        private Juwelry[] allJuwelry;
        public JuwelryContainer()
        {
            this.allJuwelry = Array.Empty<Ring>();
        }
        public JuwelryContainer(JuwelryContainer rings)
        {
            Array.Resize(ref allJuwelry, rings.Count());
            for (int i = 0; i < rings.Count(); i++)
            {
                allJuwelry[i] = rings.Get(i);
            }
        }
        /// <summary>
        /// Method to add new rings to the container
        /// </summary>
        /// <param name="ring"> new ring </param>
        public void Add(Juwelry juwelry)
        {
            Array.Resize(ref allJuwelry, allJuwelry.Length + 1);
            allJuwelry[allJuwelry.Length - 1] = juwelry;
        }
        /// <summary>
```

```
/// Method used to put ring in the given index location
/// </summary>
/// <param name="index"> index where to put the ring </param>
/// <param name="ring"> ring to be put </param>
public void Put(int index, Ring ring)
{
    allJuwelry[index] = ring;
}
/// <summary>
/// Method used to insert ring in the given index location
/// </summary>
/// <param name="index"> index where to insert the ring </param>
/// <param name="ring"> ring to be inserted </param>
public void Insert(int index, Ring ring)
{
    Array.Resize(ref allJuwelry, allJuwelry.Length + 1);
    for (int i = allJuwelry.Length; i > index; i--)
    {
        allJuwelry[i] = allJuwelry[i - 1];
    }
    allJuwelry[index] = ring;
}
/// <summary>
/// Method used to remove ring from the container
/// </summary>
/// <param name="ring"> ring to be removed </param>
public void Remove(Ring ring)
{
    int i = 0;
    int count = 0;
    foreach (Ring Rings in this.allJuwelry)
    {
        if (Rings.Equals(ring))
        {
            count++;
            for (int j = i; j < this.allJuwelry.Length - count; j++)
            {
                this.allJuwelry[j] = this.allJuwelry[j + 1];

            }
        }
        i++;
    }
    Array.Resize(ref allJuwelry, allJuwelry.Length - count);
}
/// <summary>
/// Method used to remove ring from index location in the container
/// </summary>
/// <param name="index"> index location </param>
public void RemoveAt(int index)
{
    for (int i = index; i < allJuwelry.Length; i++)
    {
        allJuwelry[i] = allJuwelry[i + 1];
    }
}
/// <summary>
/// Boolean that checks whether container has given ring
/// </summary>
/// <param name="ring"> Given ring </param>
/// <returns> boolean value </returns>
public bool Contains(Ring ring)
{
    foreach (Ring Ring in allJuwelry)
    {
```

```
                if (Ring.Equals(ring)) return true;
            }
            return false;
        }
        /// <summary>
        /// Method that sorts rings container by multiple criterias
        /// </summary>
        public void Sort()
        {
            Juwelry temp;

            for (int i = 0; i < allJuwelry.Length - 1; i++)
            {
                for (int j = i + 1; j < allJuwelry.Length; j++)
                {
                    int compare =
allJuwelry[j].Manufacturer.CompareTo(allJuwelry[i].Manufacturer);
                    if (compare > 0) continue;
                    if (compare != 0)
                    {
                        temp = allJuwelry[j];
                        allJuwelry[j] = allJuwelry[i];
                        allJuwelry[i] = temp;
                    }
                    else
                    {
                        int compareName =
allJuwelry[j].Name.CompareTo(allJuwelry[i].Name);
                        if (compareName > 0) continue;
                        if (compareName != 0)
                        {
                            temp = allJuwelry[j];
                            allJuwelry[j] = allJuwelry[i];
                            allJuwelry[i] = temp;
                        }
                        else if (allJuwelry[j].Price < allJuwelry[i].Price)
                        {
                            temp = allJuwelry[j];
                            allJuwelry[j] = allJuwelry[i];
                            allJuwelry[i] = temp;
                        }
                    }
                }
            }
        }
        /// <summary>
        /// Method used to increase container size
        /// </summary>
        /// <param name="shop"> shop container </param>
        public void AddRange(JuwelryContainer shop)
        {
            int oldLength = allJuwelry.Length;
            Array.Resize(ref allJuwelry, allJuwelry.Length +
shop.allJuwelry.Length);
            Array.Copy(shop.allJuwelry, 0, allJuwelry, oldLength,
shop.allJuwelry.Length);
        }
        /// <summary>
        /// Another method used to increase container size
        /// </summary>
        /// <param name="rings"> shop container </param>
        public void AddRange(Juwelry[] juwelries)
        {
            int oldLength = allJuwelry.Length;
            Array.Resize(ref allJuwelry, allJuwelry.Length + juwelries.Length);
```

```
            Array.Copy(juwelries, 0, allJuwelry, oldLength, juwelries.Length);
        }
        /// <summary>
        /// Method used to return how many rings are in container
        /// </summary>
        /// <returns> rings count </returns>
        public int Count()
        {
            return this.allJuwelry.Length;
        }
        /// <summary>
        /// Method used to return a ring from given location
        /// </summary>
        /// <param name="index"> given location </param>
        /// <returns> returns Jewelry </returns>
        public Juwelry Get(int index)
        {
            return this.allJuwelry[index];
        }
        /// <summary>
        /// Method to use foreach cycle
        /// </summary>
        /// <returns></returns>
        public System.Collections.IEnumerator GetEnumerator()
        {
            return allJuwelry.GetEnumerator();
        }
    }
}
```

InOut.cs

```
using System;
using System.IO;
using System.Linq;
using System.Text;

namespace U5_23
{
      class InOut
      {
            /// <summary>
            /// Method that reads data from multiple files
            /// </summary>
            /// <param name="INS"> Read files </param>
            /// <returns> Shop array with all read shops </returns>
            public static Shop[] ReadData(string[] INS)
            {
                  Shop[] allShops = Array.Empty<Shop>();
                  for(int i=0; i<INS.Length; i++)
                  {
                        Array.Resize(ref allShops, allShops.Length + 1);
                        allShops[i] = InOut.ReadJuwelry(INS[i]);
                  }
                  return allShops;
            }

            /// <summary>
            /// Method that read data from single file
            /// </summary>
            /// <param name="IN"> file to read from </param>
            /// <returns> returns a shop with jewelry from single shop </returns>
            public static Shop ReadJuwelry(string IN)
            {
                  JuwelryContainer juwelry = new JuwelryContainer();
```

```
                string[] Lines = File.ReadAllLines(IN, Encoding.UTF8);
                if (Lines.Length <= 3)
                {
                        return new Shop();
                }
                foreach (string line in Lines.Skip(3))
                {
                        string[] Values = line.Split(';');
                        if (Values.Length != 8)
                                continue;
                        string manufacturer = Values[1];
                        string title = Values[2];
                        string material = Values[3];
                        double weight = double.Parse(Values[4]);
                        int fineness = int.Parse(Values[5]);
                        double price = double.Parse(Values[6]);
                        switch (Values[0])
                        {
                                case "RING":
                                        int size = int.Parse(Values[7]);
                                        juwelry.Add(new Ring(size, manufacturer,
title, material, weight, fineness, price));
                                        break;
                                case "COLLAR":
                                        int length = int.Parse(Values[7]);
                                        juwelry.Add(new Collar(length, manufacturer,
title, material, weight, fineness, price));
                                        break;
                                case "EARRINGS":
                                        string EarringBack = Values[7];
                                        juwelry.Add(new Earrings(EarringBack,
manufacturer, title, material, weight, fineness, price));
                                        break;
                                default:
                                        break;
                        }
                }
                return new Shop(juwelry, Lines[0], Lines[1], Lines[2]);
        }
        /// <summary>
        /// Method used to print initial data in multiple files
        /// </summary>
        /// <param name="shop"></param>
        /// <param name="OUT"></param>
        public static void PrintInitialData(Shop shop, string OUT)
        {
                using (StreamWriter writer = new StreamWriter(OUT))
                {
                        if (shop.allJuwelry.Count() > 0)
                        {
                                writer.WriteLine(new string('-',
Juwelry.PrintFormat().Length));
                                writer.WriteLine(Juwelry.PrintFormat());
                                writer.WriteLine(new string('-',
Juwelry.PrintFormat().Length));


                                for (int i = 0; i < shop.allJuwelry.Count(); i++)
                                {

      writer.WriteLine(shop.allJuwelry.Get(i).ToString());
                                }
                                writer.WriteLine(new string('-', 100));
                                writer.WriteLine("");
                        }
```

```
                        else
                        {
                              writer.WriteLine("Gaminių nėra");
                        }
                  }
            }
            /// <summary>
            /// Method used to print any data to Console
            /// </summary>
            /// <param name="allJuwelry"> all juwelry </param>
            /// <param name="title"> title </param>
            public static void ShowConsole(JuwelryContainer allJuwelry, string
title)
            {
                  Console.WriteLine(title);
                  if (allJuwelry.Count() > 0)
                  {

                        Console.WriteLine(new string('-',
Juwelry.PrintFormat().Length));
                        Console.WriteLine(Juwelry.PrintFormat());
                        Console.WriteLine(new string('-',
Juwelry.PrintFormat().Length));
                        for (int i = 0; i < allJuwelry.Count(); i++)
                        {
                              Console.WriteLine(allJuwelry.Get(i).ToString());
                        }
                        Console.WriteLine(new string('-',
Juwelry.PrintFormat().Length));
                        Console.WriteLine("");
                  }
                  else
                  {
                        Console.WriteLine("No data");
                  }
            }
            public static void PrintToFile(JuwelryContainer allJuwelry, string
OUT, string title)
        {
                  string[] lines = new string[allJuwelry.Count() + 5];

                  if (allJuwelry.Count() > 0)
                  {
                        lines[0] = title;
                        lines[1] = new string('-', Juwelry.PrintFormat().Length);
                        lines[2] = Juwelry.PrintFormat();
                        lines[3] = new string('-', Juwelry.PrintFormat().Length);
                        for (int i = 0; i < allJuwelry.Count(); i++)
                        {
                              lines[i+4] = allJuwelry.Get(i).ToString();
                        }
                        lines[allJuwelry.Count()+4] = new string('-',
Juwelry.PrintFormat().Length);
                  }
                  else
                  {
                        lines[0] = title;
                        lines[1] = "No data";
                  }
                  File.WriteAllLines(OUT, lines);
            }
      }
}
```

Shop.cs

```
using System;
using System.Linq;

namespace U5_23
{
    class Shop
      {
            public JuwelryContainer allJuwelry;
            public string ShopName { get; set; }
            public string Address { get; set; }
            public string PhoneNum { get; set; }
            /// <summary>
            /// Construct for Shop
            /// </summary>
            /// <param name="juwelries"> jewelry container </param>
            /// <param name="ShopName"> name of the shop </param>
            /// <param name="Address"> adress of the shop </param>
            /// <param name="PhoneNum"> phone number of the shop </param>
            public Shop(JuwelryContainer juwelries, string ShopName, string
Address, string PhoneNum)
            {
                  this.allJuwelry = juwelries;
                  this.ShopName = ShopName;
                  this.Address = Address;
                  this.PhoneNum = PhoneNum;
            }
            /// <summary>
            /// Another constructo of Shop
            /// </summary>
            public Shop()
            {
                  allJuwelry = new JuwelryContainer();
            }
            /// <summary>
            /// Method used to add shop elements from shop array
            /// </summary>
            /// <param name="Shops"></param>
            public void Add(Shop[] Shops)
            {
                  foreach (Shop shop in Shops)
                  {
                        Add(shop);
                  }
            }
            /// <summary>
            /// Method used to add shop elements from shop
            /// </summary>
            /// <param name="shop"></param>
            public void Add(Shop shop)
            {
                  for (int i = 0; i < shop.Count(); i++)
                  {
                        Add(shop.GetJuwelry(i));
                  }
            }
            /// <summary>
            /// Method used to add jewelry
            /// </summary>
            /// <param name="juwelry"> jewelry to add</param>
            private void Add(Juwelry juwelry)
            {
                  allJuwelry.Add(juwelry);
            }
            /// <summary>
```

```
/// Method that returns count
/// </summary>
/// <returns> jewelry count </returns>
public int Count()
{
	return allJuwelry.Count();
}
/// <summary>
/// Method that returns a Jewelry at given index
/// </summary>
/// <param name="index"> given index </param>
/// <returns></returns>
public Juwelry GetJuwelry(int index)
{
	return allJuwelry.Get(index);
}
/// <summary>
/// Method that returns a Shop list with selected type of jewelry
/// </summary>
/// <param name="type"></param>
/// <returns></returns>
public Shop FilterByType(string type)
{
	Shop result = new Shop();
	for (int i = 0; i < allJuwelry.Count(); i++)
	{
		if (allJuwelry.Get(i).GetType().Name == type)
		{
			result.Add(allJuwelry.Get(i));
		}
	}
	return result;
}
/// <summary>
/// Method used to filter only the heaviest
/// </summary>
/// <returns></returns>
public JuwelryContainer FilterHeaviest()
{
	return this.FindHeaviest(allJuwelry);
}
/// <summary>
/// Method used to find heaviest jewelry
/// </summary>
/// <param name="allJuwelry"> Container with heaviest jewelry </param>
/// <returns></returns>
private JuwelryContainer FindHeaviest(JuwelryContainer allJuwelry)
{
	JuwelryContainer Heaviest = new JuwelryContainer();
	if (allJuwelry.Count() != 0)
	{
		Heaviest.Add(allJuwelry.Get(0));
	}
	for (int i = 1; i < allJuwelry.Count(); i++)
	{
		Juwelry ring = allJuwelry.Get(i);
		if (Heaviest.Get(0).GetWeight() == ring.GetWeight())
		{
			Heaviest.Add(ring);
		}
		else if (Heaviest.Get(0).GetWeight() < ring.GetWeight())
		{
			Heaviest = new JuwelryContainer();
			Heaviest.Add(ring);
		}
```

```
                }
                return Heaviest;
            }
            /// <summary>
            /// Method that finds only finest metal jewelry
            /// </summary>
            /// <param name="shopCount"></param>
            public void HighestQuality(int shopCount)
            {
                Shop Finest = new Shop();
                Quality value;
                for (int i = 0; i < allJuwelry.Count(); i++)
                {
                    Enum.TryParse(allJuwelry.Get(i).Metal, out value);
                    if ((int)value == allJuwelry.Get(i).Fineness)
                    {
                        Finest.Add(allJuwelry.Get(i));
                    }
                }
                InOut.ShowConsole(Finest.allJuwelry, String.Format("Highest
Quality from {0} shop", shopCount));
            }
            /// <summary>
            /// Method that returns jewelries that are in all shops
            /// </summary>
            /// <param name="allShops"></param>
            /// <returns></returns>
            public JuwelryContainer InAllShops(Shop[] allShops)
        {
                JuwelryContainer Filtered = new JuwelryContainer();
                foreach (Juwelry juwelry in allShops[0].allJuwelry)
                {
                    bool has = false;
                    foreach (Shop shop in allShops.Skip(1))
                    {
                        has = false;
                        foreach (Juwelry juwelry1 in shop.allJuwelry)
                        {
                            if (juwelry == juwelry1)
                            {
                                has = true;
                                break;
                            }
                        }
                        if (!has)
                            break;
                    }
                    if (has)
                        Filtered.Add(juwelry);
                }
                return Filtered;
            }
            /// <summary>
            /// Method used to filter jewelry
            /// </summary>
            /// <returns> Filtered jewelry container </returns>
            public JuwelryContainer Filtering()
        {
                JuwelryContainer Filtered = new JuwelryContainer();
                foreach(Juwelry juwelry in allJuwelry)
            {
                    if(juwelry.Price <300)
                {
                        Filtered.Add(juwelry);
                }
```

```
            }
                  return Filtered;
        }
      }
}
```

Program.cs

```
using System;
using System.IO;

namespace U5_23
{
    class Program
    {
        static void Main(string[] args)
        {
            string[] INS = { @"Shop1.csv", @"Shop2.csv", @"Shop3.csv" };
            const string Initial = @"StartDataFile.txt";
            const string OUT1 = @"Visur.csv";
            const string OUT2 = @"300.csv";
            File.Delete(OUT1);
            File.Delete(OUT2);
            Shop[] allShops = InOut.ReadData(INS);
            Shop allJuwelry = new Shop();
            allJuwelry.Add(allShops);

            InOut.PrintInitialData(allJuwelry, Initial);

            Shop AllRings = allJuwelry.FilterByType("Ring");
            InOut.ShowConsole(AllRings.FilterHeaviest(), "Heaviest rings: ");

            Shop AllEarrings = allJuwelry.FilterByType("Earrings");
            InOut.ShowConsole(AllEarrings.FilterHeaviest(), "Heaviest earrings:
");

            Shop AllCollars = allJuwelry.FilterByType("Collar");
            InOut.ShowConsole(AllCollars.FilterHeaviest(), "Heaviest collars: ");

            int shopCount = 1;
            foreach(Shop shop in allShops)
            {
                shop.HighestQuality(shopCount);
                shopCount++;
            }

            JuwelryContainer InAllShops = allShops[0].InAllShops(allShops);
            InOut.PrintToFile(InAllShops, OUT1, "Jewelry that can be found in all
shops: ");

            JuwelryContainer FilteredJuwelry = allJuwelry.Filtering();
            FilteredJuwelry.Sort();
            InOut.PrintToFile(FilteredJuwelry, OUT2, "Filtered and sorted jewelry:
");

            Console.ReadKey();
        }
    }
}
```

5.3. Pradiniai duomenys ir rezultatai

Pirmasis testas:

Pirmo testo tikslas patikrinti, ar programa ras visus sunkiausius žiedus, auskarus ir grandinėles ir ar nenustos veikti, kadangi vieno iš metalų sąraše nėra.

Shop1.csv

```
Rasa
Laisvės alėja, Kaunas
+3706291023
EARRINGS;David Yurman;KM;Silver;62;500;410;A
COLLAR;Boucheron;AG;Palladium;52;750;286;78
EARRINGS;Bvlgari;VY;Platinum;55;700;455;B
RING;Boucheron;VC;Gold;26;750;330;92
RING;Cartier;EO;Platinum;80;550;270;19
EARRINGS;David Yurman;XJ;Palladium;52;700;233;A
COLLAR;Van Cleef;YE;Palladium;33;900;358;93
RING;Harry Winston;SA;Platinum;55;850;342;32
RING;Cartier;NQ;Platinum;29;700;261;24
RING;Harry Winston;WE;Gold;17;800;494;18
EARRINGS;Buccellati;NM;Gold;96;550;411;B
EARRINGS;David Yurman;RJ;Gold;80;800;248;C
EARRINGS;Graff;WW;Platinum;70;850;210;C
RING;Tiffany;WS;Palladium;83;700;457;32
EARRINGS;Chopard;AL;Palladium;41;650;423;A
RING;David Yurman;GR;Palladium;33;550;284;28
COLLAR;Cartier;BF;Palladium;34;900;213;54
EARRINGS;Cartier;OV;Palladium;89;850;375;C
COLLAR;Van Cleef;LR;Silver;89;800;382;47
RING;Chopard;FK;Gold;10;850;310;18
RING;Chopard;WC;Silver;56;700;392;41
COLLAR;Chopard;GE;Silver;70;500;422;84
RING;Harry Winston;TZ;Palladium;48;600;236;55
RING;Buccellati;NH;Platinum;21;900;334;81
RING;Bvlgari;SX;Platinum;82;500;307;91
COLLAR;Cartier;RT;Gold;58;750;332;26
RING;Harry Winston;NL;Platinum;61;850;319;100
RING;Van Cleef;AE;Palladium;22;500;348;99
RING;Buccellati;MG;Gold;20;900;424;55
RING;Bvlgari;LG;Gold;87;650;311;74
EARRINGS;Chopard;LR;Silver;28;600;458;B
RING;Bvlgari;JC;Palladium;31;750;265;46
EARRINGS;Harry Winston;TC;Platinum;90;550;355;C
EARRINGS;Tiffany;LR;Platinum;89;550;212;A
EARRINGS;David Yurman;LB;Gold;33;900;478;C
RING;Bvlgari;DQ;Gold;20;500;339;33
RING;Tiffany;YN;Silver;27;750;397;79
EARRINGS;Graff;RU;Gold;24;650;274;A
RING;Buccellati;GY;Gold;78;900;266;32
EARRINGS;Boucheron;VP;Palladium;34;550;269;B
COLLAR;Graff;HY;Silver;21;850;480;77
EARRINGS;Tiffany;NP;Platinum;44;800;345;C
RING;Graff;HT;Gold;76;900;250;31
COLLAR;Graff;HF;Gold;97;900;215;76
EARRINGS;Van Cleef;FH;Palladium;70;550;295;C
COLLAR;Boucheron;XO;Silver;59;900;265;29
RING;Harry Winston;RU;Platinum;33;750;390;16
RING;Boucheron;FM;Palladium;58;750;355;87
```

```
COLLAR;Graff;JK;Palladium;26;500;497;100
EARRINGS;Chopard;AB;Platinum;28;800;294;A
```

Shop2.csv

```
Tyra
Savas, Kaunas
+3706291023
RING;Buccellati;LX;Palladium;16;900;456;41
RING;Boucheron;AH;Palladium;52;850;286;67
RING;Van Cleef;TC;Platinum;77;900;328;72
RING;Cartier;GJ;Platinum;35;550;220;72
EARRINGS;Graff;EF;Silver;41;650;405;A
COLLAR;Harry Winston;SF;Gold;78;800;264;24
RING;Chopard;OL;Gold;80;650;442;96
EARRINGS;David Yurman;AW;Palladium;84;750;243;C
COLLAR;Bvlgari;UQ;Silver;80;700;279;88
COLLAR;David Yurman;QP;Gold;50;650;340;18
RING;Buccellati;HM;Silver;47;900;434;58
COLLAR;Boucheron;YT;Palladium;97;650;391;83
COLLAR;Chopard;OD;Palladium;64;800;456;11
EARRINGS;Chopard;JT;Silver;10;550;243;A
RING;Cartier;IG;Silver;25;850;493;81
EARRINGS;David Yurman;XA;Silver;80;800;301;C
RING;Graff;GD;Palladium;65;700;275;93
EARRINGS;Chopard;QH;Palladium;63;550;396;A
RING;Graff;AV;Platinum;53;550;424;53
RING;Cartier;DK;Platinum;81;650;364;55
EARRINGS;David Yurman;VT;Palladium;76;600;355;B
EARRINGS;Chopard;YM;Platinum;96;750;297;A
COLLAR;Tiffany;VZ;Platinum;60;500;464;78
COLLAR;Cartier;PE;Platinum;13;550;468;56
EARRINGS;Van Cleef;CG;Platinum;69;800;310;B
RING;Buccellati;QT;Silver;39;550;424;60
EARRINGS;Chopard;ID;Palladium;45;800;400;C
EARRINGS;Harry Winston;JX;Palladium;44;850;219;B
RING;Van Cleef;RB;Gold;69;850;498;12
EARRINGS;Bvlgari;FN;Gold;79;700;363;C
EARRINGS;Bvlgari;VI;Platinum;15;500;277;C
COLLAR;Harry Winston;WX;Gold;100;550;247;13
COLLAR;Bvlgari;WY;Platinum;31;550;365;50
COLLAR;Tiffany;TE;Platinum;67;500;362;17
COLLAR;Boucheron;DI;Platinum;26;500;499;43
EARRINGS;Buccellati;YJ;Platinum;50;600;317;A
EARRINGS;Buccellati;GQ;Platinum;29;600;297;A
RING;Chopard;UH;Gold;42;900;326;29
COLLAR;Boucheron;AL;Palladium;32;500;409;72
EARRINGS;Harry Winston;NG;Platinum;68;550;458;B
COLLAR;Tiffany;GN;Silver;32;550;238;70
COLLAR;Boucheron;PA;Palladium;96;600;441;100
EARRINGS;David Yurman;EB;Silver;52;600;367;A
RING;Buccellati;XJ;Gold;26;800;236;51
COLLAR;Cartier;HL;Palladium;57;650;390;79
RING;Chopard;LI;Palladium;100;750;342;87
RING;David Yurman;AL;Platinum;76;550;449;42
RING;Graff;YG;Silver;56;700;357;55
EARRINGS;Harry Winston;PM;Gold;92;600;335;C
```

```
EARRINGS;Van Cleef;KU;Platinum;89;800;236;B
```

Shop3.csv

```
Loka
Savanorių pr. 123, Kaunas
+3706291023
EARRINGS;Bvlgari;TI;Gold;41;550;451;C
COLLAR;Van Cleef;JK;Platinum;85;750;379;49
EARRINGS;Cartier;AL;Platinum;40;550;472;A
EARRINGS;Boucheron;MU;Palladium;25;750;228;C
EARRINGS;Tiffany;ZG;Platinum;66;650;345;B
EARRINGS;Chopard;TA;Gold;34;850;323;C
COLLAR;Chopard;MT;Palladium;88;850;408;37
EARRINGS;Harry Winston;AE;Gold;77;600;304;C
COLLAR;Cartier;NS;Palladium;100;650;268;19
EARRINGS;Bvlgari;AW;Silver;25;750;394;C
COLLAR;Graff;SW;Gold;25;600;266;92
RING;Boucheron;AC;Silver;100;500;244;89
EARRINGS;Chopard;SN;Platinum;36;600;221;A
RING;Chopard;VG;Palladium;55;850;426;57
COLLAR;Boucheron;HJ;Silver;39;550;287;61
EARRINGS;Bvlgari;AO;Palladium;45;850;376;C
EARRINGS;Graff;TR;Silver;12;700;427;B
COLLAR;Buccellati;IS;Palladium;74;800;387;81
EARRINGS;Cartier;MH;Platinum;44;800;467;C
RING;Buccellati;HU;Palladium;22;500;263;34
COLLAR;Tiffany;EA;Platinum;29;500;424;59
EARRINGS;David Yurman;FO;Platinum;13;800;370;A
RING;Bvlgari;AE;Gold;36;900;411;18
COLLAR;Cartier;BE;Platinum;62;550;265;63
COLLAR;David Yurman;QS;Platinum;25;550;455;99
RING;Van Cleef;NP;Gold;100;550;357;30
RING;Graff;UE;Palladium;36;700;432;80
COLLAR;Bvlgari;AT;Palladium;41;800;317;63
RING;Graff;LU;Platinum;41;550;248;50
COLLAR;Tiffany;AX;Platinum;79;600;446;47
EARRINGS;Cartier;AL;Silver;76;800;347;B
EARRINGS;Boucheron;IZ;Silver;24;700;244;B
EARRINGS;Bvlgari;EI;Silver;25;650;430;C
COLLAR;Graff;NN;Gold;79;600;209;94
COLLAR;Tiffany;VS;Gold;89;800;315;13
RING;Boucheron;XJ;Gold;92;850;303;95
COLLAR;Bvlgari;WE;Palladium;12;800;287;69
EARRINGS;Harry Winston;IB;Gold;65;500;444;C
COLLAR;Bvlgari;HQ;Gold;43;800;279;63
EARRINGS;Van Cleef;ER;Gold;76;500;411;A
COLLAR;Boucheron;AY;Platinum;87;500;279;47
EARRINGS;Bvlgari;FG;Platinum;47;750;225;C
COLLAR;Cartier;XM;Silver;92;550;405;48
EARRINGS;Buccellati;FJ;Silver;10;900;268;B
RING;Cartier;QG;Gold;91;900;255;16
RING;Buccellati;WT;Gold;36;800;370;90
RING;Harry Winston;EY;Silver;53;750;410;15
RING;Bvlgari;CR;Silver;27;900;277;51
EARRINGS;Van Cleef;VI;Gold;54;750;271;A
COLLAR;Harry Winston;WN;Silver;95;500;477;22
```

StartDataFile.txt

Manufacturer	Name	Metal	Weight	Fineness	Price	Size	Lock	Length
David Yurman	KM	Silver	62	500	410		A	
Boucheron	AG	Palladium	52	750	286			78
Bvlgari	VY	Platinum	55	700	455		B	
Boucheron	VC	Gold	26	750	330	92		
Cartier	EO	Platinum	80	550	270	19		
David Yurman	XJ	Palladium	52	700	233		A	
Van Cleef	YE	Palladium	33	900	358			93
Harry Winston	SA	Platinum	55	850	342	32		
Cartier	NQ	Platinum	29	700	261	24		
Harry Winston	WE	Gold	17	800	494	18		
Buccellati	NM	Gold	96	550	411		B	
David Yurman	RJ	Gold	80	800	248		C	
Graff	WW	Platinum	70	850	210		C	
Tiffany	WS	Palladium	83	700	457	32		
Chopard	AL	Palladium	41	650	423		A	
David Yurman	GR	Palladium	33	550	284	28		
Cartier	BF	Palladium	34	900	213			54
Cartier	OV	Palladium	89	850	375		C	
Van Cleef	LR	Silver	89	800	382			47
Chopard	FK	Gold	10	850	310	18		
Chopard	WC	Silver	56	700	392	41		
Chopard	GE	Silver	70	500	422			84
Harry Winston	TZ	Palladium	48	600	236	55		
Buccellati	NH	Platinum	21	900	334	81		
Bvlgari	SX	Platinum	82	500	307	91		
Cartier	RT	Gold	58	750	332			26
Harry Winston	NL	Platinum	61	850	319	100		
Van Cleef	AE	Palladium	22	500	348	99		
Buccellati	MG	Gold	20	900	424	55		
Bvlgari	LG	Gold	87	650	311	74		
Chopard	LR	Silver	28	600	458		B	
Bvlgari	JC	Palladium	31	750	265	46		
Harry Winston	TC	Platinum	90	550	355		C	
Tiffany	LR	Platinum	89	550	212		A	
David Yurman	LB	Gold	33	900	478		C	
Bvlgari	DQ	Gold	20	500	339	33		
Tiffany	YN	Silver	27	750	397	79		
Graff	RU	Gold	24	650	274		A	
Buccellati	GY	Gold	78	900	266	32		
Boucheron	VP	Palladium	34	550	269		B	
Graff	HY	Silver	21	850	480			77
Tiffany	NP	Platinum	44	800	345		C	
Graff	HT	Gold	76	900	250	31		
Graff	HF	Gold	97	900	215			76
Van Cleef	FH	Palladium	70	550	295		C	
Boucheron	XO	Silver	59	900	265			29
Harry Winston	RU	Platinum	33	750	390	16		
Boucheron	FM	Palladium	58	750	355	87		
Graff	JK	Palladium	26	500	497			100
Chopard	AB	Platinum	28	800	294		A	
Buccellati	LX	Palladium	16	900	456	41		
Boucheron	AH	Palladium	52	850	286	67		
Van Cleef	TC	Platinum	77	900	328	72		
Cartier	GJ	Platinum	35	550	220	72		
Graff	EF	Silver	41	650	405		A	
Harry Winston	SF	Gold	78	800	264			24
Chopard	OL	Gold	80	650	442	96		
David Yurman	AW	Palladium	84	750	243		C	
Bvlgari	UQ	Silver	80	700	279			88
David Yurman	QP	Gold	50	650	340			18
Buccellati	HM	Silver	47	900	434	58		
Boucheron	YT	Palladium	97	650	391			83
Chopard	OD	Palladium	64	800	456			11
Chopard	JT	Silver	10	550	243		A	
Cartier	IG	Silver	25	850	493	81		

David Yurman	XA	Silver	80	800	301		C		
Graff	GD	Palladium	65	700	275	93			
Chopard	QH	Palladium	63	550	396		A		
Graff	AV	Platinum	53	550	424	53			
Cartier	DK	Platinum	81	650	364	55			
David Yurman	VT	Palladium	76	600	355		B		
Chopard	YM	Platinum	96	750	297		A		
Tiffany	VZ	Platinum	60	500	464			78	
Cartier	PE	Platinum	13	550	468			56	
Van Cleef	CG	Platinum	69	800	310		B		
Buccellati	QT	Silver	39	550	424	60			
Chopard	ID	Palladium	45	800	400		C		
Harry Winston	JX	Palladium	44	850	219		B		
Van Cleef	RB	Gold	69	850	498	12			
Bvlgari	FN	Gold	79	700	363		C		
Bvlgari	VI	Platinum	15	500	277		C		
Harry Winston	WX	Gold	100	550	247			13	
Bvlgari	WY	Platinum	31	550	365			50	
Tiffany	TE	Platinum	67	500	362			17	
Boucheron	DI	Platinum	26	500	499			43	
Buccellati	YJ	Platinum	50	600	317		A		
Buccellati	GQ	Platinum	29	600	297		A		
Chopard	UH	Gold	42	900	326	29			
Boucheron	AL	Palladium	32	500	409			72	
Harry Winston	NG	Platinum	68	550	458		B		
Tiffany	GN	Silver	32	550	238			70	
Boucheron	PA	Palladium	96	600	441			100	
David Yurman	EB	Silver	52	600	367		A		
Buccellati	XJ	Gold	26	800	236	51			
Cartier	HL	Palladium	57	650	390			79	
Chopard	LI	Palladium	100	750	342	87			
David Yurman	AL	Platinum	76	550	449	42			
Graff	YG	Silver	56	700	357	55			
Harry Winston	PM	Gold	92	600	335		C		
Van Cleef	KU	Platinum	89	800	236		B		
Bvlgari	TI	Gold	41	550	451		C		
Van Cleef	JK	Platinum	85	750	379			49	
Cartier	AL	Platinum	40	550	472		A		
Boucheron	MU	Palladium	25	750	228		C		
Tiffany	ZG	Platinum	66	650	345		B		
Chopard	TA	Gold	34	850	323		C		
Chopard	MT	Palladium	88	850	408			37	
Harry Winston	AE	Gold	77	600	304		C		
Cartier	NS	Palladium	100	650	268			19	
Bvlgari	AW	Silver	25	750	394		C		
Graff	SW	Gold	25	600	266			92	
Boucheron	AC	Silver	100	500	244	89			
Chopard	SN	Platinum	36	600	221		A		
Chopard	VG	Palladium	55	850	426	57			
Boucheron	HJ	Silver	39	550	287			61	
Bvlgari	AO	Palladium	45	850	376		C		
Graff	TR	Silver	12	700	427		B		
Buccellati	IS	Palladium	74	800	387			81	
Cartier	MH	Platinum	44	800	467		C		
Buccellati	HU	Palladium	22	500	263	34			
Tiffany	EA	Platinum	29	500	424			59	
David Yurman	FO	Platinum	13	800	370		A		
Bvlgari	AE	Gold	36	900	411	18			
Cartier	BE	Platinum	62	550	265			63	
David Yurman	QS	Platinum	25	550	455			99	
Van Cleef	NP	Gold	100	550	357	30			
Graff	UE	Palladium	36	700	432	80			
Bvlgari	AT	Palladium	41	800	317			63	
Graff	LU	Platinum	41	550	248	50			
Tiffany	AX	Platinum	79	600	446			47	
Cartier	AL	Silver	76	800	347		B		
Boucheron	IZ	Silver	24	700	244		B		

Bvlgari	EI	Silver	25	650	430		C	
Graff	NN	Gold	79	600	209			94
Tiffany	VS	Gold	89	800	315			13
Boucheron	XJ	Gold	92	850	303	95		
Bvlgari	WE	Palladium	12	800	287			69
Harry Winston	IB	Gold	65	500	444		C	
Bvlgari	HQ	Gold	43	800	279			63
Van Cleef	ER	Gold	76	500	411		A	
Boucheron	AY	Platinum	87	500	279			47
Bvlgari	FG	Platinum	47	750	225		C	
Cartier	XM	Silver	92	550	405			48
Buccellati	FJ	Silver	10	900	268		B	
Cartier	QG	Gold	91	900	255	16		
Buccellati	WT	Gold	36	800	370	90		
Harry Winston	EY	Silver	53	750	410	15		
Bvlgari	CR	Silver	27	900	277	51		
Van Cleef	VI	Gold	54	750	271		A	
Harry Winston	WN	Silver	95	500	477			22

Console langas:

```
Heaviest rings:
| Manufacturer | Name | Metal | Weight | Fineness | Price | Size | Lock | Length |
|---|---|---|---|---|---|---|---|---|
| Chopard | LI | Palladium | 100 | 750 | 342 | 87 | | |
| Boucheron | AC | Silver | 100 | 500 | 244 | 89 | | |
| Van Cleef | NP | Gold | 100 | 550 | 357 | 30 | | |

Heaviest earrings:
| Manufacturer | Name | Metal | Weight | Fineness | Price | Size | Lock | Length |
|---|---|---|---|---|---|---|---|---|
| Buccellati | NM | Gold | 96 | 550 | 411 | | B | |
| Chopard | YM | Platinum | 96 | 750 | 297 | | A | |

Heaviest collars:
| Manufacturer | Name | Metal | Weight | Fineness | Price | Size | Lock | Length |
|---|---|---|---|---|---|---|---|---|
| Harry Winston | WX | Gold | 100 | 550 | 247 | | | 13 |
| Cartier | NS | Palladium | 100 | 650 | 268 | | | 19 |

Highest Quality from 1 shop
| Manufacturer | Name | Metal | Weight | Fineness | Price | Size | Lock | Length |
|---|---|---|---|---|---|---|---|---|
| Boucheron | VC | Gold | 26 | 750 | 330 | 92 | | |
| Cartier | OV | Palladium | 89 | 850 | 375 | | C | |
| Cartier | RT | Gold | 58 | 750 | 332 | | | 26 |

Highest Quality from 2 shop
| Manufacturer | Name | Metal | Weight | Fineness | Price | Size | Lock | Length |
|---|---|---|---|---|---|---|---|---|
| Boucheron | AH | Palladium | 52 | 850 | 286 | 67 | | |
| Harry Winston | JX | Palladium | 44 | 850 | 219 | | B | |

Highest Quality from 3 shop
| Manufacturer | Name | Metal | Weight | Fineness | Price | Size | Lock | Length |
|---|---|---|---|---|---|---|---|---|
| Chopard | MT | Palladium | 88 | 850 | 408 | | | 37 |
| Chopard | VG | Palladium | 55 | 850 | 426 | 57 | | |
| Bvlgari | AO | Palladium | 45 | 850 | 376 | | C | |
| Van Cleef | VI | Gold | 54 | 750 | 271 | | A | |
```

300.csv

Filtered and sorted jewelry:

Manufacturer	Name	Metal	Weight	Fineness	Price	Size	Lock	Length
Boucheron	AC	Silver	100	500	244	89		
Boucheron	AG	Palladium	52	750	286			78
Boucheron	AH	Palladium	52	850	286	67		
Boucheron	AY	Platinum	87	500	279			47
Boucheron	HJ	Silver	39	550	287			61
Boucheron	IZ	Silver	24	700	244		B	
Boucheron	MU	Palladium	25	750	228		C	
Boucheron	VP	Palladium	34	550	269		B	
Boucheron	XO	Silver	59	900	265			29
Buccellati	FJ	Silver	10	900	268		B	
Buccellati	GQ	Platinum	29	600	297		A	
Buccellati	GY	Gold	78	900	266	32		
Buccellati	HU	Palladium	22	500	263	34		
Buccellati	XJ	Gold	26	800	236	51		
Bvlgari	CR	Silver	27	900	277	51		
Bvlgari	FG	Platinum	47	750	225		C	
Bvlgari	HQ	Gold	43	800	279			63
Bvlgari	JC	Palladium	31	750	265	46		
Bvlgari	UQ	Silver	80	700	279			88
Bvlgari	VI	Platinum	15	500	277		C	
Bvlgari	WE	Palladium	12	800	287			69
Cartier	BE	Platinum	62	550	265			63
Cartier	BF	Palladium	34	900	213			54
Cartier	EO	Platinum	80	550	270	19		
Cartier	GJ	Platinum	35	550	220	72		
Cartier	NQ	Platinum	29	700	261	24		
Cartier	NS	Palladium	100	650	268			19
Cartier	QG	Gold	91	900	255	16		
Chopard	AB	Platinum	28	800	294		A	
Chopard	JT	Silver	10	550	243		A	
Chopard	SN	Platinum	36	600	221		A	
Chopard	YM	Platinum	96	750	297		A	
David Yurman	AW	Palladium	84	750	243		C	
David Yurman	GR	Palladium	33	550	284	28		
David Yurman	RJ	Gold	80	800	248		C	
David Yurman	XJ	Palladium	52	700	233		A	
Graff	GD	Palladium	65	700	275	93		
Graff	HF	Gold	97	900	215			76
Graff	HT	Gold	76	900	250	31		
Graff	LU	Platinum	41	550	248	50		
Graff	NN	Gold	79	600	209			94
Graff	RU	Gold	24	650	274		A	
Graff	SW	Gold	25	600	266			92
Graff	WW	Platinum	70	850	210		C	
Harry Winston	JX	Palladium	44	850	219		B	
Harry Winston	SF	Gold	78	800	264			24
Harry Winston	TZ	Palladium	48	600	236	55		
Harry Winston	WX	Gold	100	550	247			13
Tiffany	GN	Silver	32	550	238			70
Tiffany	LR	Platinum	89	550	212		A	
Van Cleef	FH	Palladium	70	550	295		C	
Van Cleef	KU	Platinum	89	800	236		B	
Van Cleef	VI	Gold	54	750	271		A	

Visur.csv

Jewelry that can be found in all shops:

Manufacturer	Name	Metal	Weight	Fineness	Price	Size	Lock	Length
David Yurman	XJ	Palladium	52	700	233		A	
Chopard	AL	Palladium	41	650	423		A	

Antrasis testas:

Antro testo tikslas patikrinti, kaip programa veiks kai duomenų faile liks tik po 1 žiedą, auskarą ir grandinėlę.

Shop4.csv

```
Rasa
Laisvės alėja, Kaunas
+3706291023
EARRINGS;David Yurman;KM;Silver;62;500;410;A
```

Shop5.csv

```
Tyra
Savas, Kaunas
+3706291023
RING;Buccellati;LX;Palladium;16;900;456;41
```

Shop6.csv

```
Loka
Savanoriu pr. 123, Kaunas
+3706291023
COLLAR;Van Cleef;JK;Platinum;85;750;379;49
```

StartDataFile.txt

Manufacturer	Name	Metal	Weight	Fineness	Price	Size	Lock	Length
David Yurman	KM	Silver	62	500	410		A	
Buccellati	LX	Palladium	16	900	456	41		
Van Cleef	JK	Platinum	85	750	379			49

Console langas:

```
Heaviest rings:
| Manufacturer | Name | Metal | Weight | Fineness | Price | Size | Lock | Length |
| Buccellati | LX | Palladium | 16 | 900 | 456 | 41 | | |

Heaviest earrings:
| Manufacturer | Name | Metal | Weight | Fineness | Price | Size | Lock | Length |
| David Yurman | KM | Silver | 62 | 500 | 410 | | A | |

Heaviest collars:
| Manufacturer | Name | Metal | Weight | Fineness | Price | Size | Lock | Length |
| Van Cleef | JK | Platinum | 85 | 750 | 379 | | | 49 |

Highest Quality from 1 shop
No data
Highest Quality from 2 shop
No data
Highest Quality from 3 shop
No data
```

300.csv

```
Filtered and sorted jewelry:
No data
```

Visur.csv

```
Jewelry that can be found in all shops:
No data
```

5.4. Dėstytojo pastabos