**Условие:**  
1. Define a class that holds information about a mobile phone device: model,  
manufacturer, price, owner, battery characteristics (model, hours idle and hours  
talk) and display characteristics (size and number of colors). Define 3 separate  
classes (class GSM holding instances of the classes Battery and Display).

2. Define several constructors for the defined classes that take different sets of  
arguments (the full information for the class or part of it). Assume that model  
and manufacturer are mandatory (the others are optional). All unknown data fill with null.

3. Add an enumeration BatteryType (Li-Ion, NiMH, NiCd, …) and use it as a new field  
for the batteries.

4. Add a method in the GSM class for displaying all information about it. Try to  
override ToString().

5. Use properties to encapsulate the data fields inside the GSM, Battery and Display  
classes. Ensure all fields hold correct data at any given time.

6. Add a static field and a property IPhone4S in the GSM class to hold the  
information about iPhone 4S.

7. Write a class GSMTest to test the GSM class:  
-Create an array of few instances of the GSM class.  
-Display the information about the GSMs in the array.  
-Display the information about the static property IPhone4S.

8. Create a class Call to hold a call performed through a GSM. It should contain  
date, time, dialed phone number and duration (in seconds).

9. Add a property CallHistory in the GSM class to hold a list of the performed  
calls. Try to use the system class List.

10. Add methods in the GSM class for adding and deleting calls from the calls  
history. Add a method to clear the call history.

11. Add a method that calculates the total price of the calls in the call history.  
Assume the price per minute is fixed and is provided as a parameter.

12. Write a class GSMCallHistoryTest to test the call history functionality of the GSM class.  
-Create an instance of the GSM class.  
-Add few calls.  
-Display the information about the calls.  
-Assuming that the price per minute is 0.37 calculate and print the total  
price of the calls in the history.  
-Remove the longest call from the history   
and calculate the total price again.  
-Finally clear the call history and print it.

**Обяснение:**  
1. Създавам три класа – GSM, Battery и Display, които съдържат съответната информация, зададена в условието, със съответните полета (Fields).  
2. Създавам конструктори на класовете.  
3. Създавам изброен тип конструкция BatteryType, която съдържа три константи, и се използва като поле в класа Battery ( private BatteryType batteryModel;).  
4. В класа GSM добавям метод ToString(), който е пренаписа (override). Метода служи за принтиране на информацията от класа.  
5. Създавам свойства (Properties) в класовете GSM, Battery и Display, които служат за капсулиране на данните.  
6. Създавам статично (static) поле и свойство IPhone4S в класа GSM, за информация относно iPhone 4S.  
7. Създавам клас GSMTest, за тестване на класа GSM. Класа съдържа:  
– масив, който пази инстанция към класа GSM. Декларирам три обекта (firstGSM, secondGSM и thirdGSM), със съответните стойности, които после се присвояват от масива.  
– принтирам информация, която масива съдържа за класа GSM.  
– принтирам иформацията за Iphone4S. Тъй като той е static, не може да се достъпи чрез обек, а чрез класа GSM.  
8. Създавам клас Call, със съответните полета.  
9. В класа GSM създавам свойство CallHistory от тип Lit, което да пази информация за всички обаждания.  
10. В класа GSM създавам три метода за добавяне на обаждания (AddCalls), за изтриване на обаждане(DeleteCalls) и за изчистване на всички обаждания (ClearCalls).  
11. Създавам метод за изчисляване на цената за минута на всички обаждания.  
12. В класа GSMTest тествам класа Call.  
– създавам четвърти обект (fourthGSM).  
– добавям няколко обаждания.  
– принтирам всички обаждания.  
– изтривам едно обаждане и принтриам останалите обаждания.  
– изчислявам общата цена на всички обаждания при 0.37 за минута.  
– изчиствам всички обаждания и принтирам резултата. Логично, на конзолата не се появява нищо, което означава че метода ClearCalls работи коректно.

**class GSM.cs**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125 | using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;    namespace MobilePhoneDevice  {      class GSM      {          //Fields          private string gsmModel;          private string gsmManufacture;          private double gsmPrice;          private string gsmOwner;          private static GSM iPhone4S;            //Constructors          public GSM(string gsmModel, string gsmManufacture, double gsmPrice, string gsmOwner)          {              this.gsmModel = gsmModel;              this.gsmManufacture = gsmManufacture;              this.gsmPrice = gsmPrice;              this.gsmOwner = gsmOwner;          }            static GSM()          {              iPhone4S = new GSM("Iphone 4S", "Apple", 2000.11, "Dimitar");          }            //Properties          public string GSMModel          {              get { return gsmModel; }              set { gsmModel = value; }          }            public string GSMManufacture          {              get { return gsmManufacture; }              set { gsmManufacture = value; }          }            public double GSMPrise          {              get { return gsmPrice; }              set { gsmPrice = value; }          }            public string GSMOwner          {              get { return gsmOwner; }              set { gsmOwner = value; }          }            public static GSM IPhone4S          {              get { return iPhone4S; }              set { iPhone4S = value; }          }            public List<Call> CallHistory = new List<Call>();            //Methods          public override string ToString()          {              Console.WriteLine("---------- GSM ----------");              Console.WriteLine("GSM model: " + this.gsmModel);              Console.WriteLine("GSM manufacture: " + this.gsmManufacture);              Console.WriteLine("GSM price: {0} $", this.gsmPrice);              Console.WriteLine("GSM owner: " + this.gsmOwner);              Console.WriteLine();              return base.ToString();          }            public void AddCalls(DateTime dateAndTime, string dialedPhoneNumber, int duration)          {              Call call = new Call(dateAndTime, dialedPhoneNumber, duration);              CallHistory.Add(call);          }            public void DeleteCalls(int duration)          {              for (int i = 0; i < CallHistory.Count; i++)              {                  if (CallHistory[i].Duration == duration)                  {                      CallHistory.RemoveAt(i);                      i--;                  }              }          }            public void ClearCalls()          {              CallHistory.Clear();          }            public void CalculateTotalPrice(double pricePerMinute)          {              double wholeTime = 0;              for (int i = 0; i < CallHistory.Count; i++)              {                  wholeTime += CallHistory[i].Duration;              }                double price = pricePerMinute \* (Math.Ceiling(wholeTime / 60));              Console.WriteLine("Total price: " + price);              Console.WriteLine();          }            public void PrintCalls()          {              for (int i = 0; i < CallHistory.Count; i++)              {                  Console.WriteLine("---------- Calls ----------");                  Console.WriteLine("Date and Time: " + CallHistory[i].DateAndTime);                  Console.WriteLine("Dialed phone number: " + CallHistory[i].DialedPhoneNumber);                  Console.WriteLine("Duration: {0} seconds ", CallHistory[i].Duration);                  Console.WriteLine();              }          }      }  } |

**class Battery.cs**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48 | using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;    namespace MobilePhoneDevice  {      class Battery      {          //Enumeration         public enum BatteryType          {              LiIon, NiMH, NiCd          }            //Fields          private BatteryType batteryModel;          private int batteryHoursIdle;          private int batteryHoursTalk;            //Constructors          public Battery(BatteryType batteryModel, int batteryHoursIdle, int batteryHoursTalk)          {              this.batteryModel = batteryModel;              this.batteryHoursIdle = batteryHoursIdle;              this.batteryHoursTalk = batteryHoursTalk;          }            //Properties          public BatteryType BatteryModel          {              get { return batteryModel; }          }            public int BatteryHoursIdle          {              get { return batteryHoursIdle; }              set { batteryHoursIdle = value; }          }            public int BatteryHoursTalk          {              get { return batteryHoursTalk; }              set { batteryHoursTalk = value; }          }      }  } |

**class Display.cs**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35 | using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;    namespace MobilePhoneDevice  {      class Display      {          //Fields          private double displaySize;          private string displayColors;            //Constructors          public Display(double displaySize, string displayColors)          {              this.displaySize = displaySize;              this.displayColors = displayColors;          }            //Properties          public double DisplaySize          {              get { return displaySize; }              set { displaySize = value; }          }            public string DisplayColors          {              get { return displayColors; }              set { displayColors = value; }          }      }  } |

**class GSMTest.cs**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56 | using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;    namespace MobilePhoneDevice  {      class GSMTest      {          static void Main()          {              GSM[] array = new GSM[3];                GSM firstGSM = new GSM("Galaxy S 4", "Samsung", 800.00, "Misho");              array[0] = firstGSM;                GSM secondGSM = new GSM("Xperia", "Sony", 700.01, "Ivan");              array[1] = secondGSM;                GSM thirdGSM = new GSM("Optimus", "LG", 500.21, "Daniel");              array[2] = thirdGSM;                //Print information about the GSMs in the array              for (int i = 0; i < array.Length; i++)              {                  array[i].ToString();              }                //Print information about the static property IPhone4S              GSM.IPhone4S.ToString();                //CallHistory test              GSM fourthGSM = new GSM("Desire 500", "HTC", 1111, "Maria");                //Add calls and print              Console.WriteLine("Add some calls and print...");              fourthGSM.AddCalls(DateTime.Now, "088088311", 55);              fourthGSM.AddCalls(DateTime.Now, "088088088", 94);              fourthGSM.AddCalls(DateTime.Now, "088111221", 33);              fourthGSM.PrintCalls();                //Delete call and print              Console.WriteLine("Delete call and print... ");              fourthGSM.DeleteCalls(55);              fourthGSM.PrintCalls();                //Calculate total price              fourthGSM.CalculateTotalPrice(0.37);                //Clear calls and print              fourthGSM.ClearCalls();              fourthGSM.PrintCalls();          }      }  } |

**class Call.cs**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43 | using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;    namespace MobilePhoneDevice  {      class Call      {          //Fields          private DateTime dateAndTime;          private string dialedPhoneNumber;          private int duration;            //Constructors          public Call(DateTime dateTime, string dialedPhoneNumber, int duration)          {              this.dateAndTime = dateTime;              this.dialedPhoneNumber = dialedPhoneNumber;              this.duration = duration;          }            //Properties          public DateTime DateAndTime          {              get { return dateAndTime; }              set { dateAndTime = value; }          }            public string DialedPhoneNumber          {              get { return dialedPhoneNumber; }              set { dialedPhoneNumber = value; }          }            public int Duration          {              get { return duration; }              set { duration = value; }          }      }  } |