**Звіт**

**Залікова робота**

**з дисципліни «Веб-орієнтована розробка програмного забезпечення С#»**

**Зробив: Ярцев Д.А.**

**Варіант 9.**

Магазин годинників. В БД зберігається інформація про годинники різних типів та марок, що продаються в магазині.

Для годинників необхідно зберігати:

- марку;

- тип (кварцові, механічні);

- ціну;

- кількість;

- реквізити виробника.

Для виробників необхідно зберігати:

- назву;

- країну.

Завдання:

- Вивести всі марки для заданого типу годинників.

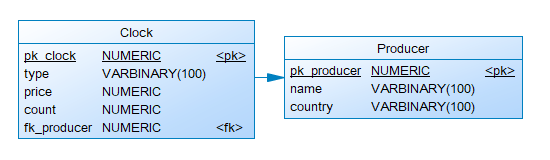
- Вивести інформацію про механічні годинники, ціна на які не перевищує задану.

- Вивести марки годинників, які виготовлені в заданій країні.

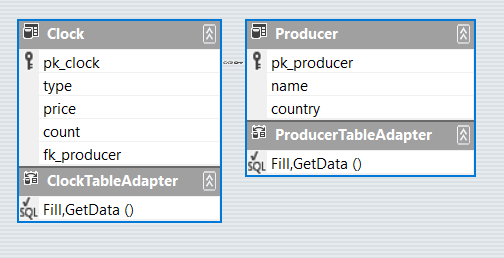
- Вивести виробників, загальна сума годинників яких є в наявності в магазині не перевищує задану.

**Хід роботи**

В якості бази даних буде використано локальну базу даних local MS SQL в console application, що було створено за допомогою Visual Studio 2014

Маємо наступну фізичну модель даних: 

Створили в схему даних в local MS SQL



Та наповнив таблиці даними:

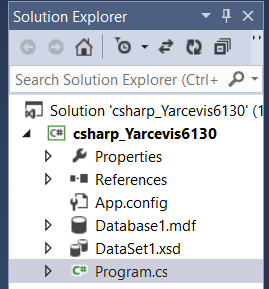
|  |  |  |
| --- | --- | --- |
| *0* | *Producer\_1* | *country\_5* |
| *1* | *Producer\_2* | *country\_2* |
| *2* | *Producer\_3* | *country\_1* |
| *3* | *Producer\_4* | *country\_2* |
| *4* | *Producer\_5* | *country\_5* |
| *5* | *Producer\_6* | *country\_4* |
| *6* | *Producer\_7* | *country\_2* |
| *7* | *Producer\_8* | *country\_5* |
| *8* | *Producer\_9* | *country\_2* |
| *9* | *Producer\_10* | *country\_2* |
| *10* | *Producer\_11* | *country\_5* |
| *11* | *Producer\_12* | *country\_5* |
| *12* | *Producer\_13* | *country\_2* |
| *13* | *Producer\_14* | *country\_3* |
| *14* | *Producer\_15* | *country\_5* |
| *15* | *Producer\_16* | *country\_3* |
| *16* | *Producer\_17* | *country\_2* |
| *17* | *Producer\_18* | *country\_1* |
| *18* | *Producer\_19* | *country\_5* |
| *19* | *Producer\_20* | *country\_5* |
| *20* | *Producer\_21* | *country\_3* |
| *21* | *Producer\_22* | *country\_1* |
| *22* | *Producer\_23* | *country\_5* |
| *23* | *Producer\_24* | *country\_2* |
| *24* | *Producer\_25* | *country\_4* |
| *25* | *Producer\_26* | *country\_2* |
| *26* | *Producer\_27* | *country\_2* |
| *27* | *Producer\_28* | *country\_5* |
| *28* | *Producer\_29* | *country\_3* |
| *29* | *Producer\_30* | *country\_2* |
| *30* | *Producer\_31* | *country\_3* |
| *31* | *Producer\_32* | *country\_1* |
| *32* | *Producer\_33* | *country\_5* |
| *33* | *Producer\_34* | *country\_1* |
| *NULL* | *NULL* | *NULL* |

*Таблиця «Propucer»*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *0* | *type\_3* | *7666* | *965* | *23* | *mark\_*10 |
| *1* | *mechanical* | *2181* | *271* | *31* | *mark\_*11 |
| *2* | *type\_3* | *3037* | *634* | *3* | *mark\_*1 |
| *3* | *mechanical* | *9841* | *555* | *10* | *mark\_*2 |
| *4* | *type\_5* | *5798* | *501* | *5* | *mark\_*7 |
| *5* | *type\_6* | *2260* | *527* | *25* | *mark\_*8 |
| *6* | *type\_7* | *7846* | *722* | *7* | *mark\_*9 |
| *7* | *type\_8* | *9996* | *169* | *1* | *mark\_*8 |
| *8* | *type\_9* | *8641* | *894* | *10* | *mark\_*9 |
| *9* | *type\_10* | *9040* | *373* | *20* | *mark\_*10 |
| *10* | *type\_1* | *4442* | *948* | *30* | *mark\_*11 |
| *11* | *type\_2* | *9284* | *508* | *12* | *mark\_*1 |
| *12* | *type\_3* | *7745* | *464* | *7* | *mark\_*2 |
| *13* | *mechanical* | *4738* | *225* | *13* | *mark\_*3 |
| *14* | *type\_5* | *1711* | *214* | *16* | *mark\_*4 |
| *15* | *type\_6* | *5469* | *290* | *5* | *mark\_*9 |
| *16* | *type\_7* | *3818* | *723* | *20* | *mark\_*8 |
| *17* | *mechanical* | *7203* | *860* | *16* | *mark\_*9 |
| *18* | *type\_5* | *2219* | *608* | *27* | *mark\_*9 |
| *19* | *type\_6* | *3474* | *569* | *16* | *mark\_*8 |
| *20* | *type\_7* | *2751* | *422* | *7* | *mark\_*9 |
| *21* | *type\_8* | *4738* | *489* | *20* | *mark\_*10 |
| *22* | *type\_9* | *1379* | *602* | *33* | *mark\_*11 |
| *23* | *type\_10* | *5142* | *444* | *5* | *mark\_*1 |
| *24* | *type\_1* | *3595* | *705* | *9* | *mark\_*2 |
| *25* | *type\_2* | *9253* | *732* | *29* | *mark\_*3 |
| *26* | *type\_3* | *4837* | *921* | *25* | *mark\_*10 |
| *27* | *mechanical* | *6571* | *541* | *10* | *mark\_*11 |
| *28* | *type\_5* | *1819* | *241* | *5* | *mark\_*1 |
| *29* | *type\_6* | *1628* | *149* | *10* | *mark\_*2 |
| *30* | *type\_7* | *8319* | *879* | *17* | *mark\_*3 |
| *31* | *type\_6* | *7146* | *587* | *32* | *mark\_*3 |
| *32* | *type\_3* | *7925* | *403* | *14* | *mark\_*4 |
| *33* | *mechanical* | *4096* | *401* | *10* | *mark\_*5 |
| *34* | *type\_5* | *2441* | *120* | *12* | *mark\_*6 |
| *35* | *type\_6* | *1436* | *5* | *32* | *mark\_*7 |
| *36* | *type\_7* | *9128* | *733* | *2* | *mark\_*8 |
| *37* | *type\_8* | *5400* | *521* | *14* | *mark\_*1 |
| *38* | *type\_9* | *5003* | *673* | *23* | *mark\_*2 |
| *39* | *type\_10* | *3548* | *495* | *32* | *mark\_*3 |
| *40* | *type\_1* | *6209* | *171* | *9* | *mark\_*4 |
| *41* | *type\_2* | *1880* | *489* | *26* | *mark\_*1 |
| *42* | *type\_3* | *8893* | *872* | *14* | *mark\_*2 |
| *43* | *mechanical* | *3956* | *481* | *2* | *mark\_*3 |
| *44* | *type\_5* | *6828* | *916* | *31* | *mark\_*4 |
| *45* | *type\_6* | *5747* | *101* | *10* | *mark\_*5 |
| *46* | *type\_7* | *3182* | *744* | *4* | *mark\_*1 |
| *47* | *type\_8* | *4658* | *973* | *28* | *mark\_*2 |
| *48* | *type\_3* | *2136* | *375* | *22* | *mark\_*3 |
| *49* | *mechanical* | *9164* | *713* | *27* | *mark\_*4 |
| *50* | *type\_5* | *4413* | *501* | *26* | *mark\_*5 |
| *51* | *type\_6* | *1370* | *447* | *0* | *mark\_*6 |
| *52* | *type\_7* | *6504* | *750* | *1* | *mark\_*1 |
| *53* | *type\_8* | *8154* | *67* | *27* | *mark\_*2 |
| *54* | *type\_9* | *9686* | *228* | *9* | *mark\_*3 |
| *55* | *type\_10* | *5455* | *402* | *15* | *mark\_*4 |
| *56* | *type\_1* | *3860* | *41* | *33* | *mark\_*5 |
| *57* | *type\_2* | *1203* | *138* | *27* | *mark\_*6 |
| *58* | *type\_3* | *2877* | *50* | *18* | *mark\_*7 |
| *59* | *mechanical* | *6324* | *661* | *23* | *mark\_*8 |
| *60* | *type\_5* | *7632* | *627* | *7* | *mark\_*1 |
| *61* | *type\_6* | *8906* | *819* | *8* | *mark\_*2 |
| *62* | *type\_7* | *6385* | *735* | *16* | *mark\_*3 |

*Таблиця «Clock»*

Після створення додатка маємо такі файли:



**роботa програми**

zachot variant 9 Yarcev DA is6130

================================================================================

Выберите действие:

1 - Вивести вс? марки для заданого типу годинник?в

2 - Вивести ?нформац?ю про механ?чн? годинники, ц?на на як? не перевищує задану

3 - Вивести марки годинник?в, як? виготовлен? в задан?й країн?

4 - Вивести виробник?в, загальна сума годинник?в яких є в наявност? в магазин? не перевищує задану

5 - завершити програму

№ действия:1

--------------------------------------------------------------------------------

Пожалуйства введите наименование типа (значения в диапазоне[type\_1 - type\_10]): type\_1

[mark]

--------------------------------------------------------------------------------

[mark\_11]

[mark\_2]

[mark\_4]

[mark\_5]

================================================================================

Выберите действие:

1 - Вивести вс? марки для заданого типу годинник?в

2 - Вивести ?нформац?ю про механ?чн? годинники, ц?на на як? не перевищує задану

3 - Вивести марки годинник?в, як? виготовлен? в задан?й країн?

4 - Вивести виробник?в, загальна сума годинник?в яких є в наявност? в магазин? не перевищує задану

5 - завершити програму

№ действия:2

--------------------------------------------------------------------------------

Механические часы

Пожалуйства введите максимальную цену( диапазон [1000;10000]): 5000

[pk\_clock] [type] [price] [count] [fk\_producer] [mark]

--------------------------------------------------------------------------------

[0] [type\_3] [2852] [831] [11] [mark\_10]

[7] [type\_8] [1696] [486] [2] [mark\_8]

[9] [type\_10] [4018] [840] [14] [mark\_10]

[10] [type\_1] [2756] [927] [19] [mark\_11]

[13] [mechanical] [1461] [584] [6] [mark\_3]

[14] [type\_5] [3443] [906] [30] [mark\_4]

[15] [type\_6] [2385] [395] [31] [mark\_9]

[17] [mechanical] [3765] [894] [17] [mark\_9]

[20] [type\_7] [2727] [498] [18] [mark\_9]

[25] [type\_2] [3197] [524] [0] [mark\_3]

[29] [type\_6] [1413] [796] [12] [mark\_2]

[30] [type\_7] [3708] [461] [23] [mark\_3]

[31] [type\_6] [4693] [200] [0] [mark\_3]

[34] [type\_5] [3760] [832] [2] [mark\_6]

[37] [type\_8] [4438] [229] [30] [mark\_1]

[42] [type\_3] [4545] [19] [32] [mark\_2]

[43] [mechanical] [3285] [736] [2] [mark\_3]

[44] [type\_5] [3757] [900] [18] [mark\_4]

[46] [type\_7] [1295] [750] [25] [mark\_1]

[47] [type\_8] [4432] [152] [25] [mark\_2]

[50] [type\_5] [4520] [944] [23] [mark\_5]

[51] [type\_6] [3260] [596] [32] [mark\_6]

[53] [type\_8] [4652] [266] [16] [mark\_2]

[55] [type\_10] [3185] [735] [8] [mark\_4]

[60] [type\_5] [3923] [708] [26] [mark\_1]

================================================================================

Выберите действие:

1 - Вивести вс? марки для заданого типу годинник?в

2 - Вивести ?нформац?ю про механ?чн? годинники, ц?на на як? не перевищує задану

3 - Вивести марки годинник?в, як? виготовлен? в задан?й країн?

4 - Вивести виробник?в, загальна сума годинник?в яких є в наявност? в магазин? не перевищує задану

5 - завершити програму

№ действия:3

--------------------------------------------------------------------------------

[country]

--------------------------------------------------------------------------------

[country\_5]

[country\_2]

[country\_1]

[country\_2]

[country\_5]

[country\_4]

[country\_2]

[country\_5]

[country\_2]

[country\_2]

[country\_5]

[country\_5]

[country\_2]

[country\_3]

[country\_5]

[country\_3]

[country\_2]

[country\_1]

[country\_5]

[country\_5]

[country\_3]

[country\_1]

[country\_5]

[country\_2]

[country\_4]

[country\_2]

[country\_2]

[country\_5]

[country\_3]

[country\_2]

[country\_3]

[country\_1]

[country\_5]

[country\_1]

--------------------------------------------------------------------------------

марки часов в заданной стране

введите название страны (из таблицы выше): country\_10

[pk\_clock] [type] [price] [count] [fk\_producer] [mark] [pk\_producer] [name] [country]

--------------------------------------------------------------------------------

================================================================================

Выберите действие:

1 - Вивести вс? марки для заданого типу годинник?в

2 - Вивести ?нформац?ю про механ?чн? годинники, ц?на на як? не перевищує задану

3 - Вивести марки годинник?в, як? виготовлен? в задан?й країн?

4 - Вивести виробник?в, загальна сума годинник?в яких є в наявност? в магазин? не перевищує задану

5 - завершити програму

№ действия:3

--------------------------------------------------------------------------------

[country]

--------------------------------------------------------------------------------

[country\_5]

[country\_2]

[country\_1]

[country\_2]

[country\_5]

[country\_4]

[country\_2]

[country\_5]

[country\_2]

[country\_2]

[country\_5]

[country\_5]

[country\_2]

[country\_3]

[country\_5]

[country\_3]

[country\_2]

[country\_1]

[country\_5]

[country\_5]

[country\_3]

[country\_1]

[country\_5]

[country\_2]

[country\_4]

[country\_2]

[country\_2]

[country\_5]

[country\_3]

[country\_2]

[country\_3]

[country\_1]

[country\_5]

[country\_1]

--------------------------------------------------------------------------------

марки часов в заданной стране

введите название страны (из таблицы выше): country\_5

[pk\_clock] [type] [price] [count] [fk\_producer] [mark] [pk\_producer] [name] [country]

--------------------------------------------------------------------------------

[0] [type\_3] [2852] [831] [11] [mark\_10] [11] [Producer\_12] [country\_5]

[2] [type\_3] [6990] [871] [0] [mark\_1] [0] [Producer\_1] [country\_5]

[3] [mechanical] [9355] [690] [32] [mark\_2] [32] [Producer\_33] [country\_5]

[9] [type\_10] [4018] [840] [14] [mark\_10] [14] [Producer\_15] [country\_5]

[10] [type\_1] [2756] [927] [19] [mark\_11] [19] [Producer\_20] [country\_5]

[18] [type\_5] [6990] [852] [22] [mark\_9] [22] [Producer\_23] [country\_5]

[19] [type\_6] [7787] [567] [18] [mark\_8] [18] [Producer\_19] [country\_5]

[20] [type\_7] [2727] [498] [18] [mark\_9] [18] [Producer\_19] [country\_5]

[24] [type\_1] [5102] [804] [7] [mark\_2] [7] [Producer\_8] [country\_5]

[25] [type\_2] [3197] [524] [0] [mark\_3] [0] [Producer\_1] [country\_5]

[27] [mechanical] [5622] [850] [22] [mark\_11] [22] [Producer\_23] [country\_5]

[31] [type\_6] [4693] [200] [0] [mark\_3] [0] [Producer\_1] [country\_5]

[40] [type\_1] [8670] [258] [7] [mark\_4] [7] [Producer\_8] [country\_5]

[41] [type\_2] [9287] [238] [27] [mark\_1] [27] [Producer\_28] [country\_5]

[42] [type\_3] [4545] [19] [32] [mark\_2] [32] [Producer\_33] [country\_5]

[44] [type\_5] [3757] [900] [18] [mark\_4] [18] [Producer\_19] [country\_5]

[45] [type\_6] [8020] [244] [4] [mark\_5] [4] [Producer\_5] [country\_5]

[49] [mechanical] [9003] [637] [27] [mark\_4] [27] [Producer\_28] [country\_5]

[51] [type\_6] [3260] [596] [32] [mark\_6] [32] [Producer\_33] [country\_5]

[57] [type\_2] [9965] [538] [14] [mark\_6] [14] [Producer\_15] [country\_5]

[58] [type\_3] [5575] [165] [4] [mark\_7] [4] [Producer\_5] [country\_5]

[59] [mechanical] [7163] [195] [19] [mark\_8] [19] [Producer\_20] [country\_5]

================================================================================

Выберите действие:

1 - Вивести вс? марки для заданого типу годинник?в

2 - Вивести ?нформац?ю про механ?чн? годинники, ц?на на як? не перевищує задану

3 - Вивести марки годинник?в, як? виготовлен? в задан?й країн?

4 - Вивести виробник?в, загальна сума годинник?в яких є в наявност? в магазин? не перевищує задану

5 - завершити програму

№ действия:4

--------------------------------------------------------------------------------

Механические часы

Пожалуйства введите максимальную цену( диапазон [1000;infinite]): 5000

[name] [sum\_price]

--------------------------------------------------------------------------------

[Producer\_12] [2852]

[Producer\_18] [3765]

[Producer\_32] [2385]

[Producer\_7] [1461]

================================================================================

Выберите действие:

1 - Вивести вс? марки для заданого типу годинник?в

2 - Вивести ?нформац?ю про механ?чн? годинники, ц?на на як? не перевищує задану

3 - Вивести марки годинник?в, як? виготовлен? в задан?й країн?

4 - Вивести виробник?в, загальна сума годинник?в яких є в наявност? в магазин? не перевищує задану

5 - завершити програму

№ действия:5

--------------------------------------------------------------------------------

================================================================================

Program is ending...

**Код програми**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data;

using System.Data.SqlClient;

using System.Data.OleDb;

using System.Data.Common;

namespace csharp\_Yarcevis6130

{

class Program

{

public static string constr =

@"Data Source=(LocalDB)\v11.0;AttachDbFilename=|DataDirectory|Database1.mdf;Integrated Security=True";

static void select\_task1()

{

SqlConnection Conn = new SqlConnection(Program.constr);

Conn.Open();

DataSet ds = new DataSet();

Console.Write(" Пожалуйства введите наименование типа (значения в диапазоне[type\_1 - type\_10]): ");

string word = Console.ReadLine();

string commStr = "SELECT distinct mark FROM Clock where type ='"+word+"'";

SqlDataAdapter adapter = new SqlDataAdapter(commStr, Conn);

adapter.MissingSchemaAction = System.Data.MissingSchemaAction.AddWithKey;

adapter.Fill(ds);

DataTable table = ds.Tables[0];

foreach (DataColumn column in table.Columns)

{

Console.Write("[" + column.ColumnName + "] ");

}

Console.WriteLine("\n" + new string('-', 80));

foreach (DataRow row in table.Rows)

{

foreach (DataColumn column in table.Columns)

{

Console.Write("[" + row[column] + "] ");

}

Console.WriteLine();

}

Conn.Close();

}

static void select()

{

SqlConnection Conn = new SqlConnection(Program.constr);

Conn.Open();

DataSet ds = new DataSet();

string commStr = "SELECT country FROM Producer";

SqlDataAdapter adapter = new SqlDataAdapter(commStr, Conn);

adapter.MissingSchemaAction = System.Data.MissingSchemaAction.AddWithKey;

adapter.Fill(ds);

DataTable table = ds.Tables[0];

foreach (DataColumn column in table.Columns)

{

Console.Write("[" + column.ColumnName + "] ");

}

Console.WriteLine("\n" + new string('-', 80));

foreach (DataRow row in table.Rows)

{

foreach (DataColumn column in table.Columns)

{

Console.Write("[" + row[column] + "] ");

}

Console.WriteLine();

}

Conn.Close();

}

static void select\_task2() {

SqlConnection Conn = new SqlConnection(Program.constr);

Conn.Open();

DataSet ds = new DataSet();

Console.WriteLine(" Механические часы");

Console.Write(" Пожалуйства введите максимальную цену( диапазон [1000;10000]): ");

string word = Console.ReadLine();

string commStr = "SELECT \* FROM Clock where price <='" + word + "'";

SqlDataAdapter adapter = new SqlDataAdapter(commStr, Conn);

adapter.MissingSchemaAction = System.Data.MissingSchemaAction.AddWithKey;

adapter.Fill(ds);

DataTable table = ds.Tables[0];

foreach (DataColumn column in table.Columns)

{

Console.Write("[" + column.ColumnName + "] ");

}

Console.WriteLine("\n" + new string('-', 80));

foreach (DataRow row in table.Rows)

{

foreach (DataColumn column in table.Columns)

{

Console.Write("[" + row[column] + "] ");

}

Console.WriteLine();

}

Conn.Close();

}

static void select\_task3() {

SqlConnection Conn = new SqlConnection(Program.constr);

Conn.Open();

DataSet ds = new DataSet();

Console.WriteLine(" марки часов в заданной стране");

Console.Write(" введите название страны (из таблицы выше): ");

string word = Console.ReadLine();

string commStr = "select \* FROM Clock a1 inner join Producer a2 on a1.fk\_producer = a2.pk\_producer where country = '"+word+"'";

SqlDataAdapter adapter = new SqlDataAdapter(commStr, Conn);

adapter.MissingSchemaAction = System.Data.MissingSchemaAction.AddWithKey;

adapter.Fill(ds);

DataTable table = ds.Tables[0];

foreach (DataColumn column in table.Columns)

{

Console.Write("[" + column.ColumnName + "] ");

}

Console.WriteLine("\n" + new string('-', 80));

foreach (DataRow row in table.Rows)

{

foreach (DataColumn column in table.Columns)

{

Console.Write("[" + row[column] + "] ");

}

Console.WriteLine();

}

Conn.Close();

}

static void select\_task4() {

SqlConnection Conn = new SqlConnection(Program.constr);

Conn.Open();

DataSet ds = new DataSet();

Console.WriteLine(" Механические часы");

Console.Write(" Пожалуйства введите максимальную цену( диапазон [1000;infinite]): ");

string word = Console.ReadLine();

string commStr = "select name, sum(price) as sum\_price FROM Clock a1 inner join Producer a2 on a1.fk\_producer = a2.pk\_producer group by name having sum(price) <= "+word+";";

SqlDataAdapter adapter = new SqlDataAdapter(commStr, Conn);

adapter.MissingSchemaAction = System.Data.MissingSchemaAction.AddWithKey;

adapter.Fill(ds);

DataTable table = ds.Tables[0];

foreach (DataColumn column in table.Columns)

{

Console.Write("[" + column.ColumnName + "] ");

}

Console.WriteLine("\n" + new string('-', 80));

foreach (DataRow row in table.Rows)

{

foreach (DataColumn column in table.Columns)

{

Console.Write("[" + row[column] + "] ");

}

Console.WriteLine();

}

Conn.Close();

}

static void lineCh(char s)

{

for (int i = 0; i < 80; i++)

Console.Write(s);

Console.WriteLine();

}

static void Main(string[] args)

{

Console.WriteLine("zachot variant 9 Yarcev DA is6130");

lineCh('=');

//управляющий алгоритм

{

int i = 0;

bool key = true;

while (key)

{

bool key\_input = true;

while (key\_input)

{

key\_input = false;

Console.WriteLine(" Выберите действие:");

Console.WriteLine(" 1 - Вивести всі марки для заданого типу годинників");

Console.WriteLine(" 2 - Вивести інформацію про механічні годинники, ціна на які не перевищує задану");

Console.WriteLine(" 3 - Вивести марки годинників, які виготовлені в заданій країні");

Console.WriteLine(" 4 - Вивести виробників, загальна сума годинників яких є в наявності в магазині не перевищує задану");

Console.WriteLine(" 5 - завершити програму");

Console.Write(" № действия:");

i = Int32.Parse(Console.ReadLine());

if (i < 1 || i > 5)

{

Console.WriteLine(" Не верно, повторите ввод.");

key\_input = true;

}

}

lineCh('-');

switch (i)

{

case 1: select\_task1(); break;

case 2: select\_task2(); break;

case 3:

select();

lineCh('-');

select\_task3(); break;

case 4: select\_task4(); break;

case 5: key = false; break;

}

lineCh('=');

}

}

Console.WriteLine("Program is ending...");

Console.ReadLine();

}

}

}

**Висновок**

Дана робота дозволила мені примінити отримані знання на практиці