

Перовод байтів в біт, кбайт, мбайт, гбайт і навпаки

1. Для визначеного варіанту завдання реалізувати діаграми USE CASE, Activity, Class, Components у відповідності до обраної мови програмування та середовища розробки.

Діаграма: Use Case

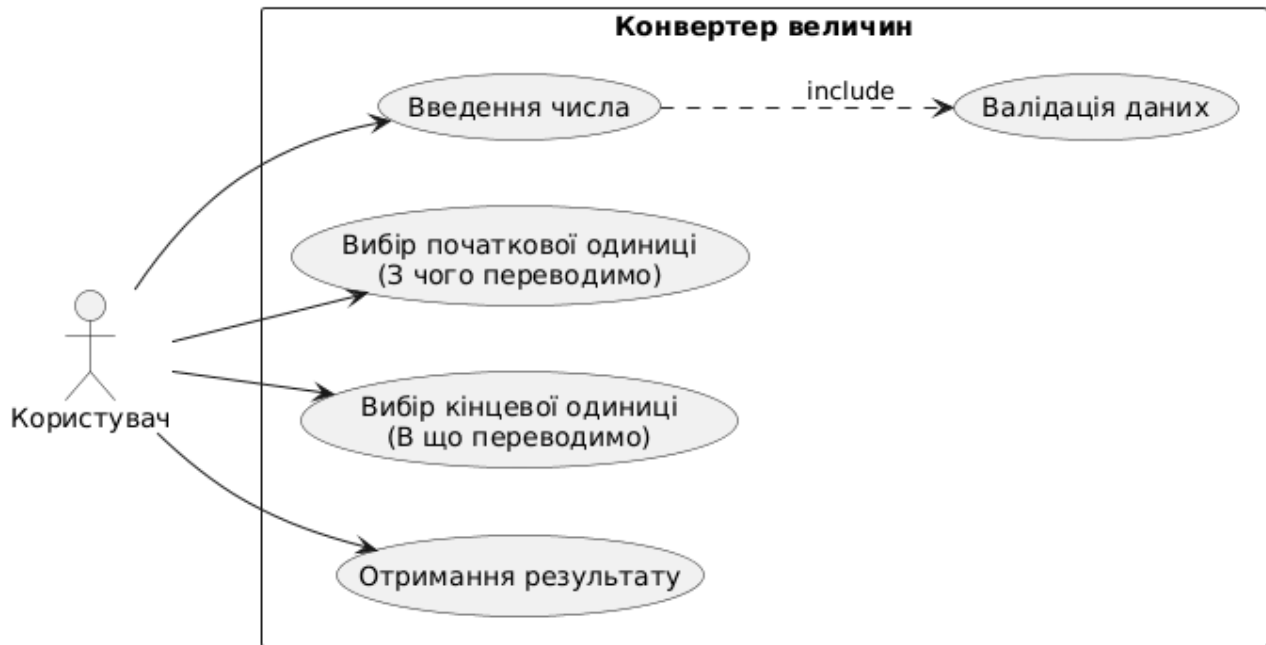


Рис.1. Діаграма Use Case

Лістинг:

```
@startuml
left to right direction
skinparam packageStyle rectangle
actor "Користувач" as user
package "Конвертер величин" {
    usecase "Введення числа" as UC1
    usecase "Вибір початкової одиниці\n(З чого переводимо)" as UC2
    usecase "Вибір кінцевої одиниці\n(В що переводимо)" as UC3
    usecase "Отримання результату" as UC4
    usecase "Валідація даних" as UC5
}
user --> UC1
user --> UC2
user --> UC3
user --> UC4
UC1 ..> UC5 : include
@enduml
```

## Діаграма: Activity



Рис.2. Діаграма Activity

## Лістинг:

```
@startuml
start
:Користувач вводить число;
:Користувач обирає "З чого" (напр. МБ);
:Користувач обирає "В що" (напр. Біт);
if (Число коректне?) then (Ні)
:Показати повідомлення про помилку;
stop
else (Так)
:Конвертація вхідного числа у Байти\n(Базова величина);
:Створення об'єкта ByteConverter;
:Виклик методу конвертації\n(ToBits, ToKiloBytes тощо)\nzалежно від вибору "В що"
:Виведення одного результату на екран;
endif
stop
@enduml
```

## Діаграма: Class

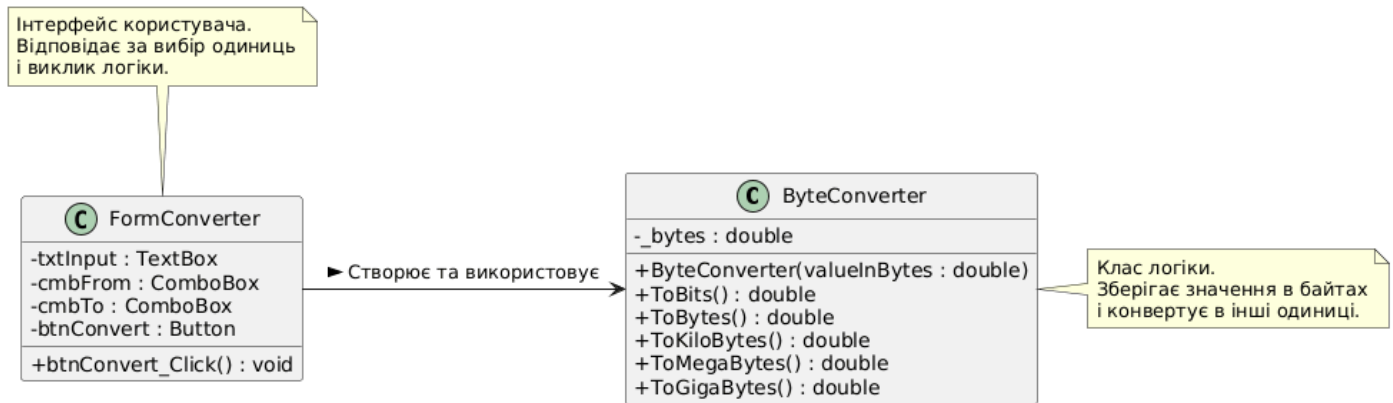


Рис.3. Діаграма Class

## Лістинг:

```

@startuml
skinparam classAttributeIconSize 0
class "ByteConverter" {
    - _bytes : double
    + ByteConverter(valueInBytes : double)
    + ToBits() : double
    + ToBytes() : double
    + ToKiloBytes() : double
    + ToMegaBytes() : double
    + ToGigaBytes() : double
}
note right of ByteConverter : Клас логіки.\nЗберігає значення в байтах\nі конвертує в інші одиниці.
class "FormConverter" {
    - txtInput : TextBox
    - cmbFrom : ComboBox
    - cmbTo : ComboBox
    - btnConvert : Button
    + btnConvert_Click() : void
}
note top of FormConverter : Інтерфейс користувача.\nВідповідає за вибір одиниць\nі виклик логіки.
FormConverter -> ByteConverter : Створює та використовує >
@enduml
  
```

## Діаграма: Component

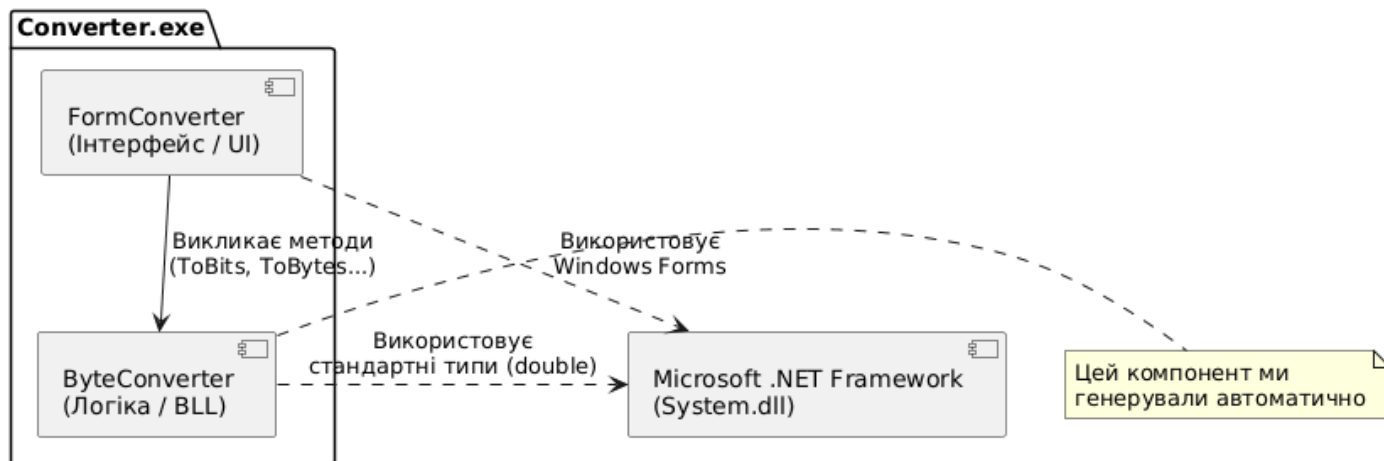


Рис.4. Діаграма Component

## Лістинг:

```
@startuml
skinparam componentStyle uml2
package "Converter.exe" {
    component [FormConverter\n(Інтерфейс / UI)] as GUI
    component [ByteConverter\n(Логіка / BLL)] as Logic
}
component "Microsoft .NET Framework\n(System.dll)" as DotNet
GUI -down-> Logic : Викликає методи\n(ToBits, ToBytes...)
GUI .> DotNet : Використовує\nWindows Forms
Logic .> DotNet : Використовує\nстандартні типи (double)
note right of Logic
    Цей компонент ми
    генерували автоматично
end note
@enduml
```

2. Виконати кодогенерацію діаграми класів в середовище розробки та реалізувати програмний продукт. - <http://msdn.microsoft.com/ruru/library/ff657795.aspx>

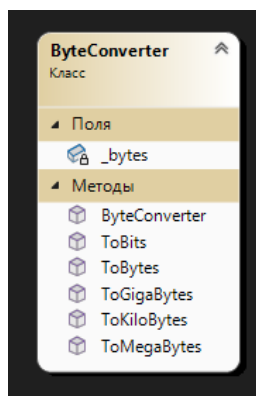


Рис. 5. Клас ByteConverter

## Згенерований лістинг:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Converter
{
    public class ByteConverter
    {
        private double _bytes;

        public ByteConverter()
        {
            throw new NotImplementedException();
        }

        public double ToBits()
        {
            throw new NotImplementedException();
        }

        public double ToKiloBytes()
        {
            throw new NotImplementedException();
        }

        public double ToMegaBytes()
        {
            throw new NotImplementedException();
        }

        public double ToGigaBytes()
        {
            throw new NotImplementedException();
        }

        public double ToBytes()
        {
            throw new NotImplementedException();
        }
    }
}
```

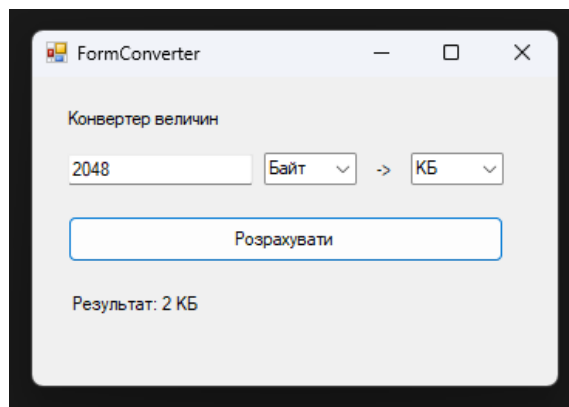


Рис. 6. Інтерфейс програми

## Лістинг:

```
using System;

namespace Converter
{
    public class ByteConverter
    {
        private double _bytes;

        public ByteConverter(double valueInBytes)
        {
            this._bytes = valueInBytes;
        }

        public double ToBits()
        {
            return _bytes * 8.0;
        }

        public double ToBytes()
```

```
{  
    return _bytes;  
}  
  
public double ToKiloBytes()  
{  
    return _bytes / 1024.0;  
}  
  
public double ToMegaBytes()  
{  
    return _bytes / (1024.0 * 1024.0);  
}  
  
public double ToGigaBytes()  
{  
    return _bytes / (1024.0 * 1024.0 * 1024.0);  
}  
}
```

Посилання на гіт: <https://github.com/VadymLeus/Y4S1-SWMA>