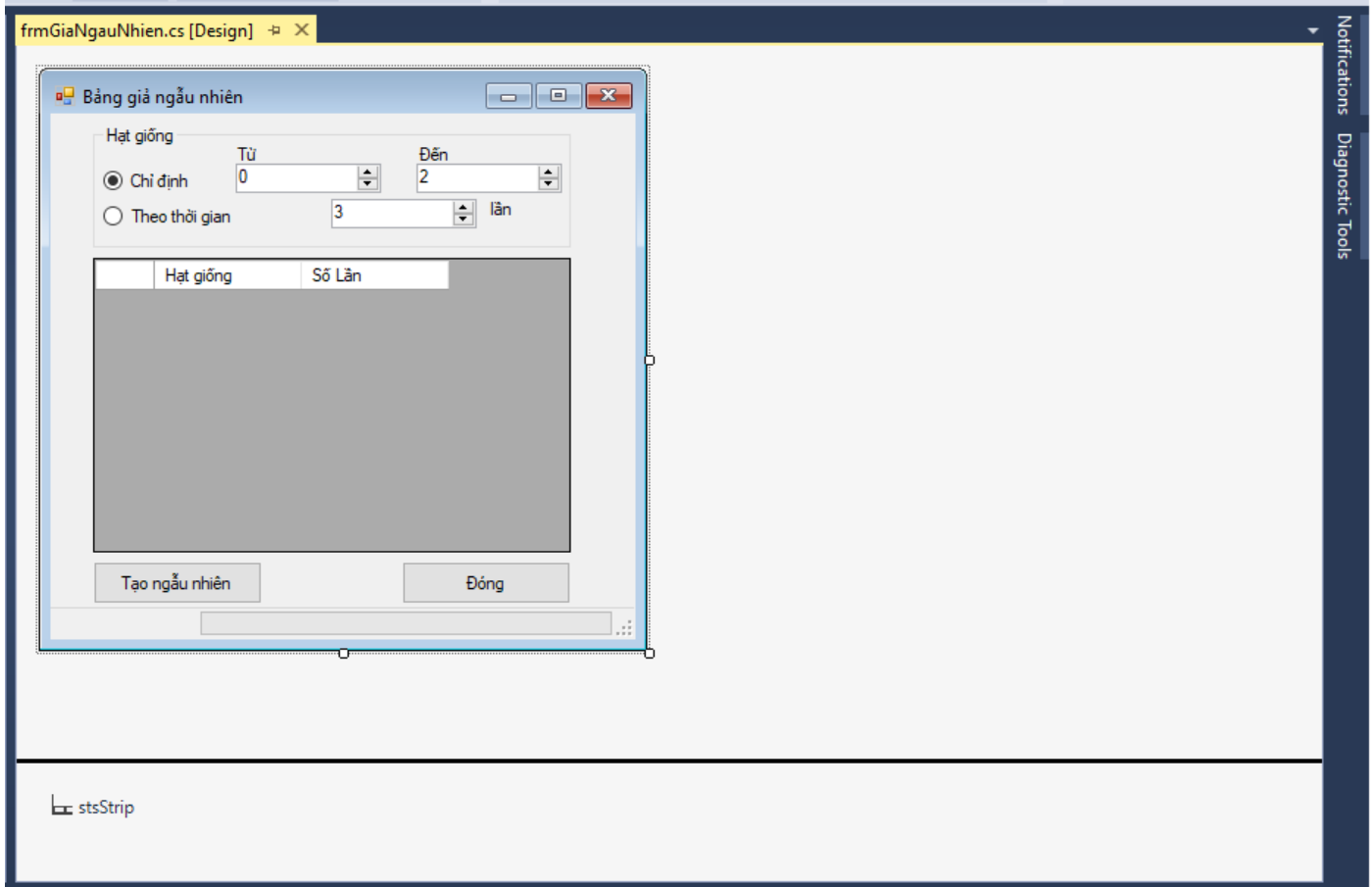


Bài thực hành 3 – Bảo mật hệ thống thông tin Giả Ngẫu Nhiên (Pseudo Random)

1. Anh/chị tạo một đề án C# mới, có tên GiaNgauNhiên.
2. Anh/chị sửa lại Form1.cs thành frmGiaNgauNhiên.cs, đồng thời sửa lại thành phần Text của form này thành Bảng Giả Ngẫu nhiên.
3. Anh/chị thiết kế giao diện của frmGiaNgauNhiên như sau:



Kích cỡ của frmGiaNgauNhiên là 414, 396.

STT	Tên đối tượng	Loại đối tượng	Text	Enabled	Vị trí
1	gbHatGiong	GroupBox	Hạt giống		Dòng đầu tiên
2	dgBangSoLan	DataGrid			Ngay dưới gbHatGiong
3	btnTaoNgauNhiên	Button	Tạo ngẫu nhiên	True	Ngay dưới dgBangSoLan
4	btnDong	Button	Đóng	True	Bên phải btnTaoNgauNhiên
5	stsStrip	StatusStrip	Trạng thái		Dòng cuối cùng

4. Các đối tượng bên trong gbHatGiong

STT	Tên đối tượng	Loại đối tượng	Text	Các thuộc tính khác	Vị trí
1	lblTu	Label	Từ		Dòng đầu tiên
2	lblDen	Label	Đến		Bên phải lblTu
3	rbChiDinh	RadioButton	Chỉ định	Checked=True	Cột đầu tiên, cách ra 1 dòng ỏ trên
4	nudHatGiongTu	Numeric UpDown		Value = 0 Minimum = -100 Maximum = 100	Bên phải rdChiDinh và ngay dưới lblTu
5	nudHatGiongDen	Numeric UpDown		Value = 2 Minumum = -100	Bên phải nudHatGiongTu

				Maxmum = 100	
6	rbThoiGian	RadioButton	Theo thời gian	Checked=False	Ngay dưới rbChiDinh
7	nudSoLan	Numeric UpDown		Value = 3 Minumum = 0 Maximum = 100	Bên phải rbThoiGian
8	lblLan	Label	lần		Bên phải nudSoLan

5. Đối tượng dgBangSoLan có các thuộc tính:

- AllowUserToAddRows = False
- Columns có các cột như sau:

STT	Tên đối tượng	Loại đối tượng	HeaderText	DataPropertyName	DefaultCellStyle
1	dgcHatGiong	DataGridView TextBoxColumn	Hạt giống	HatGiong	Alignment=MiddleRight
2	dgcSoLan	DataGridView TextBoxColumn	Số lần	SoLan	Alignment=MiddleRight Format=N0

6. Đối tượng stsStrip có các thuộc tính:

STT	Tên đối tượng	Loại đối tượng	AutoSize	Size
1	stsTrangThai	ToolStripStatusLabel	False	100,17
2	stsTienDo	ToolStripProgressBar	False	280,16

7. Anh/chị khai báo các biến sau:

- Biến riêng tư thứ nhất, có tên dsBangSoLan, với kiểu DataSet, và được khởi tạo bằng hàm tạo không tham biến.
- Biến riêng tư thứ hai, có tên dtBangSoLan, với kiểu DataTable, và được khởi tạo bằng hàm tạo không tham biến.

8. Anh/chị khai báo các phương thức riêng tư như sau:

- Phương thức riêng tư thứ nhất có tên TimSoLan. Phương thức này có một tham biến hình thức, có tên HatGiong, với kiểu số nguyên. Phương thức này trả về kiểu long, tức là số nguyên dài. Nhiệm vụ của phương thức này là:

```
Random RndObj = new Random(HatGiong);
```

```
int SoNgauNhanDauTien = RndObj.Next();
```

```
int SoNgauNhan;
```

```
long SoLan = 0;
```

```
do
```

```
{
```

```
    SoNgauNhan = RndObj.Next();
```

```
    SoLan++;
```

```
} while (SoNgauNhan != SoNgauNhanDauTien);
```

```
return SoLan;
```

- Phương thức riêng tư thứ hai, có tên ThemVaoBang. Phương thức này có hai tham biến hình thức. Tham biến hình thức thứ nhất có tên HatGiong, với kiểu số nguyên. Tham biến hình thức thứ hai, có tên SoLan với kiểu long. Phương thức này không có kiểu dữ liệu trả về. Nhiệm vụ của phương thức này là:

```
dtBangSoLan.Rows.Add(HatGiong, SoLan);
```

- Phương thức riêng tư thứ ba, có tên KiemHopLe. Phương thức này không có tham biến hình thức. Phương thức này không có kiểu dữ liệu trả về. Nhiệm vụ của phương thức này là:

```
btnTaoNgauNhan.Enabled = rbChiDinh.Checked && (nudHatGiongTu.Value <= nudHatGiongDen.Value)
|| rbThoiGian.Checked && nudSoLan.Value > 0;
```

9. Phương thức xử lý biến cố Load của frmGiaNgauNhan, có tên frmGiaNgauNhan_Load. Phương thức này có hai tham biến hình thức. Tham biến hình thức thứ nhất có tên sender, với kiểu object. Tham biến hình thức thứ hai có tên e, với kiểu EventArgs. Phương thức này không có kiểu dữ liệu trả về. Nhiệm vụ của phương thức này là:

```
dtBangSoLan.Columns.Add(new DataColumn("HatGiong", System.Type.GetType("System.Int32")));
dtBangSoLan.Columns.Add(new DataColumn("SoLan", System.Type.GetType("System.Decimal")));
dtBangSoLan.TableName = "BangSo";
```

```
dsBangSoLan.Tables.Add(dtBangSoLan);  
dgBangSoLan.DataSource = dsBangSoLan.Tables["BangSo"]; // dtBangSoLan;
```

10. Phương thức xử lý biến cố CheckedChanged của rbChiDinh là rbChiDinh_CheckedChanged. Phương thức này có hai tham biến hình thức. Tham biến hình thức thứ nhất có tên sender, với kiểu object. Tham biến hình thức thứ hai có tên e, với kiểu EventArgs. Phương thức này không có kiểu dữ liệu trả về. Nhiệm vụ của phương thức này là:

```
nudHatGiongTu.Enabled = true;  
nudHatGiongDen.Enabled = true;  
nudSoLan.Enabled = false;  
KiemHopLe();
```

11. Phương thức xử lý biến cố CheckedChanged của rbThoiGian là rbThoiGian_CheckedChanged. Phương thức này có hai tham biến hình thức. Tham biến hình thức thứ nhất có tên sender, với kiểu object. Tham biến hình thức thứ hai có tên e, với kiểu EventArgs. Phương thức này không có kiểu dữ liệu trả về. Nhiệm vụ của phương thức này là:

```
nudHatGiongTu.Enabled = false;  
nudHatGiongDen.Enabled = false;  
nudSoLan.Enabled = true;  
KiemHopLe();
```

12. Phương thức xử lý biến cố ValueChanged của cả ba nudHatGiongTu, nudHatGiongDen và nudSoLan đều có chung một trình xử lý nud_ValueChanged. Phương thức này có hai tham biến hình thức. Tham biến hình thức thứ nhất có tên sender, với kiểu object. Tham biến hình thức thứ hai có tên e, với kiểu EventArgs. Phương thức này không có kiểu dữ liệu trả về. Nhiệm vụ của phương thức này là:

```
KiemHopLe();
```

13. Phương thức xử lý biến cố Click của btnTaoNgauNhien là btnTaoNgauNhien_Click. Phương thức này có hai tham biến hình thức. Tham biến hình thức thứ nhất có tên sender, với kiểu object. Tham biến hình thức thứ hai có tên e, với kiểu EventArgs. Phương thức này không có kiểu dữ liệu trả về. Nhiệm vụ của phương thức này là:

```
btnTaoNgauNhien.Enabled = false;  
stsStrip.Items[0].Text = "Đang tính toán ...";  
stsTienDo.Value = 0;  
  
if (rbChiDinh.Checked)  
{  
    int HatGiongTu = (int)nudHatGiongTu.Value;  
    int HatGiongDen = (int)nudHatGiongDen.Value;  
    dtBangSoLan.Rows.Clear();  
    Application.DoEvents();  
    stsTienDo.Step = 100 / (HatGiongDen - HatGiongTu + 1);  
  
    for (int HatGiong = HatGiongTu; HatGiong <= HatGiongDen; HatGiong++)  
    {  
        stsTienDo.Value = (HatGiong - HatGiongTu + 1) * 100 / (HatGiongDen - HatGiongTu + 1);  
        long SoLan = TimSoLan(HatGiong);  
        ThemVaoBang(HatGiong, SoLan);  
        Application.DoEvents();  
    }  
}  
else  
{  
    dtBangSoLan.Rows.Clear();  
    Application.DoEvents();  
    int SoLanChay = (int)nudSoLan.Value;  
    stsTienDo.Step = 100 / SoLanChay;  
  
    for (int lan=0; lan<(int) nudSoLan.Value; lan++)  
    {  
        stsTienDo.Value = (lan + 1) * 100 / SoLanChay;
```

```

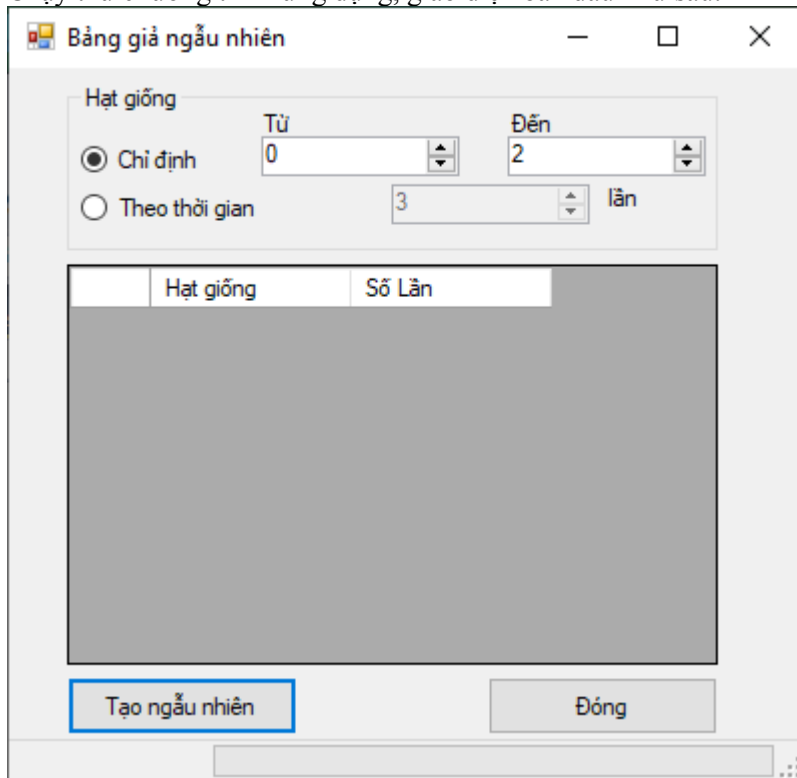
        int HatGiong = (int)DateTime.Now.Millisecond;
        long SoLan = TimSoLan(HatGiong);
        ThemVaoBang(HatGiong, SoLan);
        Application.DoEvents();
    }
}
stsTienDo.Value = 100;
stsStrip.Items[0].Text = "Đã tính xong";
btnTaoNgauNhiem.Enabled = true;
}

```

14. Phương thức xử lý biến cố Click của btnDong là btnDong_Click. Phương thức này có hai tham biến hình thức. Tham biến hình thức thứ nhất có tên sender, với kiểu object. Tham biến hình thức thứ hai có tên e, với kiểu EventArgs. Phương thức này không có kiểu dữ liệu trả về. Nhiệm vụ của phương thức này là:

```
this.Close();
```

15. Chạy thử chương trình ứng dụng, giao diện ban đầu như sau:



16. Bấm nút btnTaoNgauNhiem, ta có kết quả:

Bảng giả ngẫu nhiên

Hạt giống

☒ Chỉ định Từ 0 Đến 2

☐ Theo thời gian 3 lần

	Hạt giống	Số Lần
▶	0	720.515.461

Tạo ngẫu nhiên Đóng

Đang tính toán ...

Bảng giả ngẫu nhiên

Hạt giống

☒ Chỉ định Từ 0 Đến 2

☐ Theo thời gian 3 lần

	Hạt giống	Số Lần
▶	0	720.515.461
	1	168.564.517
	2	657.973.803

Tạo ngẫu nhiên Đóng

Đã tính xong

17. Chọn nút bấm tròn rbThoiGian như sau:

Bảng giả ngẫu nhiên

Hạt giống

☐ Chỉ định Từ 0 Đến 2

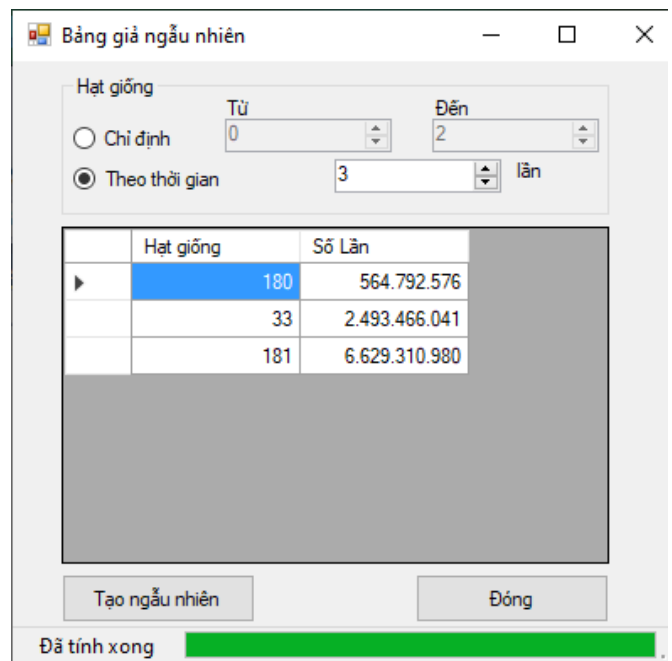
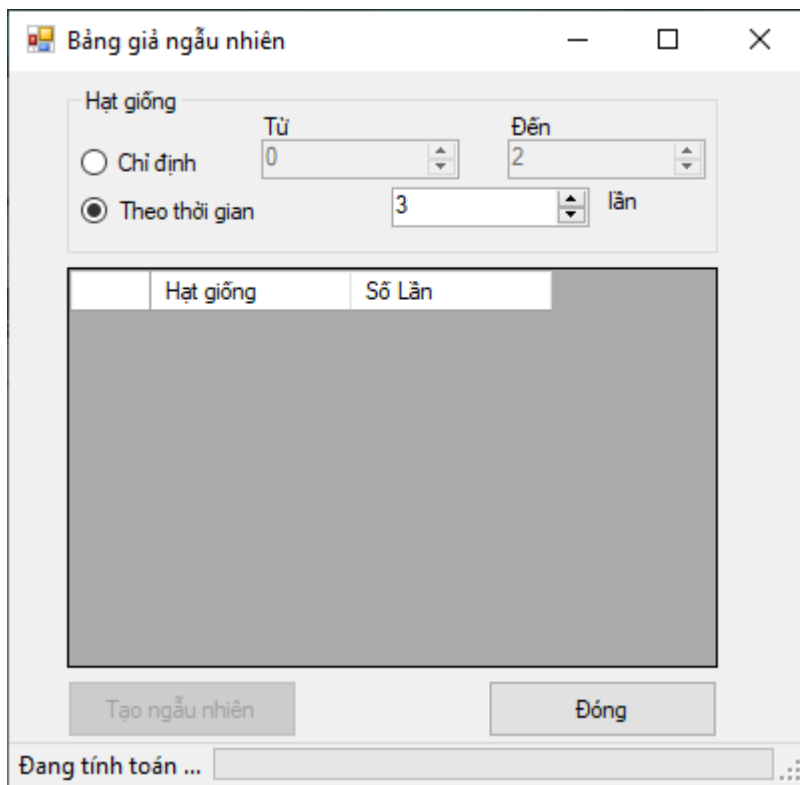
☒ Theo thời gian 3 lần

	Hạt giống	Số Lần
▶	0	720.515.461
	1	168.564.517
	2	657.973.803

Tạo ngẫu nhiên Đóng

Đã tính xong

18. Sau đó bấm btnTaoNgauNhiên, ta có kết quả:



Bảng giả ngẫu nhiên

Hạt giống

☒ Chỉ định
 Từ Đến

☐ Theo thời gian
 lần

	Hạt giống	Số Lần
▶	0	720.515.461
	1	168.564.517
	2	657.973.803
	3	2.910.403.007
	4	2.253.746.860
	5	1.120.524.335

Tạo ngẫu nhiên Đóng

Đã tính xong

20. Với trường hợp rbChiDinh, anh/chị hãy đối chiếu với kết quả:

<div>×</div> <p>Hạt giống 0, Số lần = 720515461</p> <div>OK</div>	<div>×</div> <p>Hạt giống 1, Số lần = 168564517</p> <div>OK</div>	<div>×</div> <p>Hạt giống 2, Số lần = 657973803</p> <div>OK</div>
<div>×</div> <p>Hạt giống 3, Số lần = 2910403007</p> <div>OK</div>	<div>×</div> <p>Hạt giống 4, Số lần = 2253746860</p> <div>OK</div>	<div>×</div> <p>Hạt giống 5, Số lần = 1120524335</p> <div>OK</div>
<div>×</div> <p>Hạt giống 6, Số lần = 1193831975</p> <div>OK</div>	<div>×</div> <p>Hạt giống 7, Số lần = 2503854432</p> <div>OK</div>	<div>×</div> <p>Hạt giống 8, Số lần = 337809903</p> <div>OK</div>
<div>×</div> <p>Hạt giống 9, Số lần = 8900619474</p> <div>OK</div>	<div>×</div> <p>Hạt giống 10, Số lần = 4474196029</p> <div>OK</div>	<div>×</div> <p>Hạt giống 11, Số lần = 3632107379</p> <div>OK</div>

21. Với lần chọn rbThoiGian, mỗi lần đều có hạt giống ngẫu nhiên theo thời gian (tính bằng mili-giây):

Bảng giả ngẫu nhiên

Hạt giống

☐ Chỉ định Từ 0 Đến 2

☒ Theo thời gian 3 lần

	Hạt giống	Số Lần
▶	33	2.493.466.041
	895	105.332.467
	780	1.038.847.186

Tạo ngẫu nhiên Đóng

Đã tính xong

22. Với kết quả 10 lần lấy hạt giống ngẫu nhiên theo thời gian, ta có:

Bảng giả ngẫu nhiên

Hạt giống

☐ Chỉ định Từ 0 Đến 2

☒ Theo thời gian 10 lần

	Hạt giống	Số Lần
▶	198	8.803.103.493

Tạo ngẫu nhiên Đóng

Đang tính toán ...

Bảng giả ngẫu nhiên

Hạt giống

☐ Chỉ định Từ 0 Đến 2

☒ Theo thời gian 10 lần

	Hạt giống	Số Lần
▶	198	8.803.103.493
	894	596.609.863
	219	551.830.530
	963	956.437.569

Tạo ngẫu nhiên Đóng

Đang tính toán ...

Bảng giả ngẫu nhiên

Hạt giống

☐ Chỉ định

Từ 0 Đến 2

☒ Theo thời gian

10 lần

	Hạt giống	Số Lần
▶	198	8.803.103.493
	894	596.609.863
	219	551.830.530
	963	956.437.569
	230	861.959.795
	947	1.806.311.641

Tạo ngẫu nhiên Đóng

Đang tính toán ...

Bảng giả ngẫu nhiên

Hạt giống

☐ Chỉ định

Từ 0 Đến 2

☒ Theo thời gian

10 lần

	Hạt giống	Số Lần
	219	551.830.530
	963	956.437.569
	230	861.959.795
	947	1.806.311.641
	47	3.134.996.186
	816	2.146.216.604
	660	1.077.016.141
	555	891.759.536

Tạo ngẫu nhiên Đóng

Đã tính xong

23. Bấm nút btnDong để đóng lại chương trình ứng dụng.

24. Chúc các bạn thành công.

---oOo---