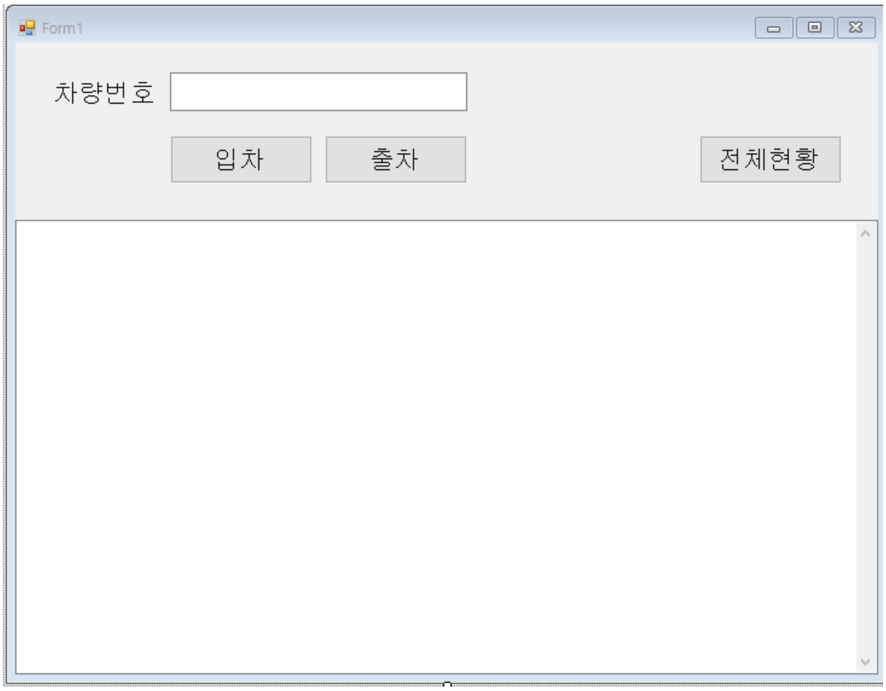
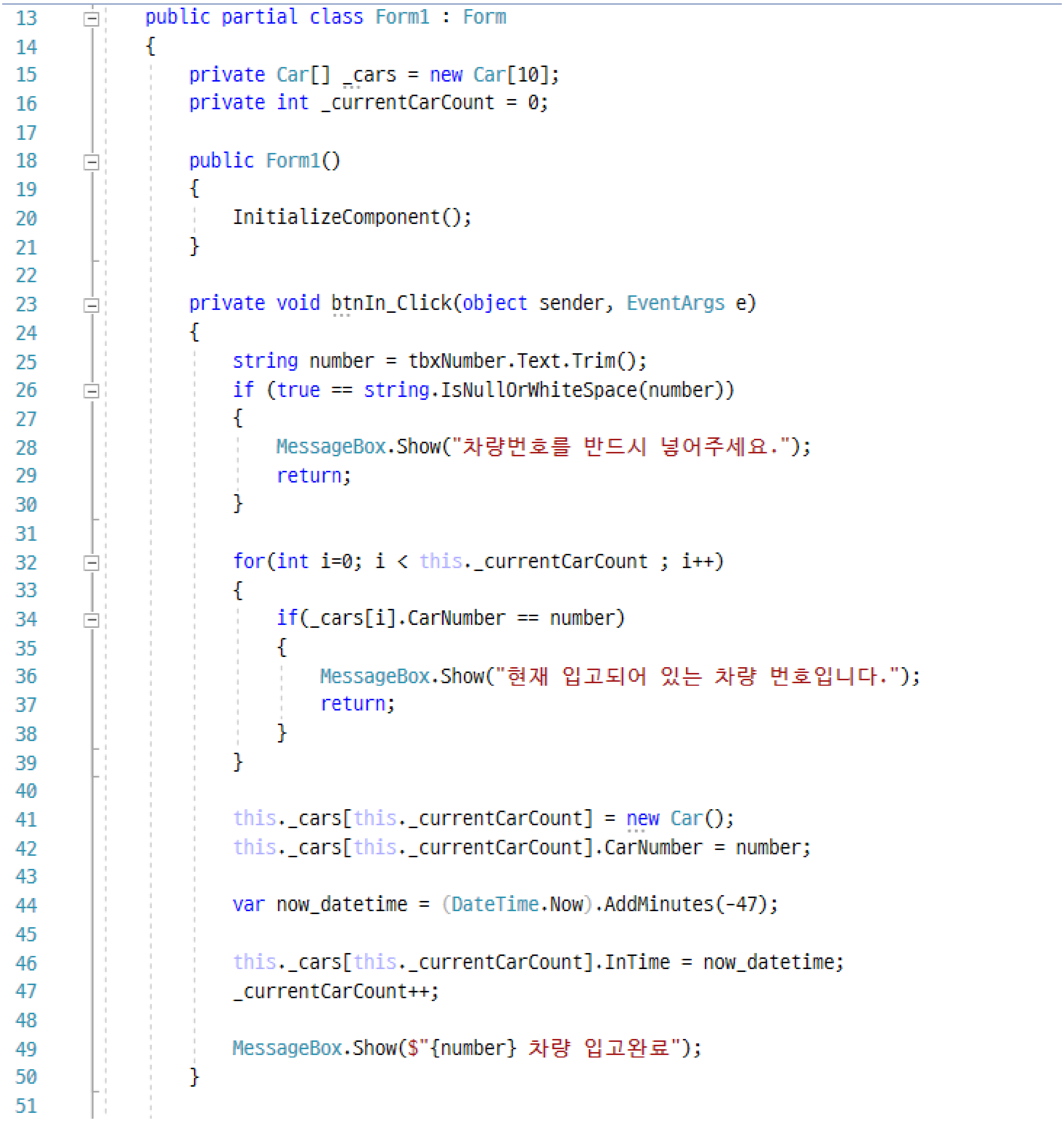
[화면]

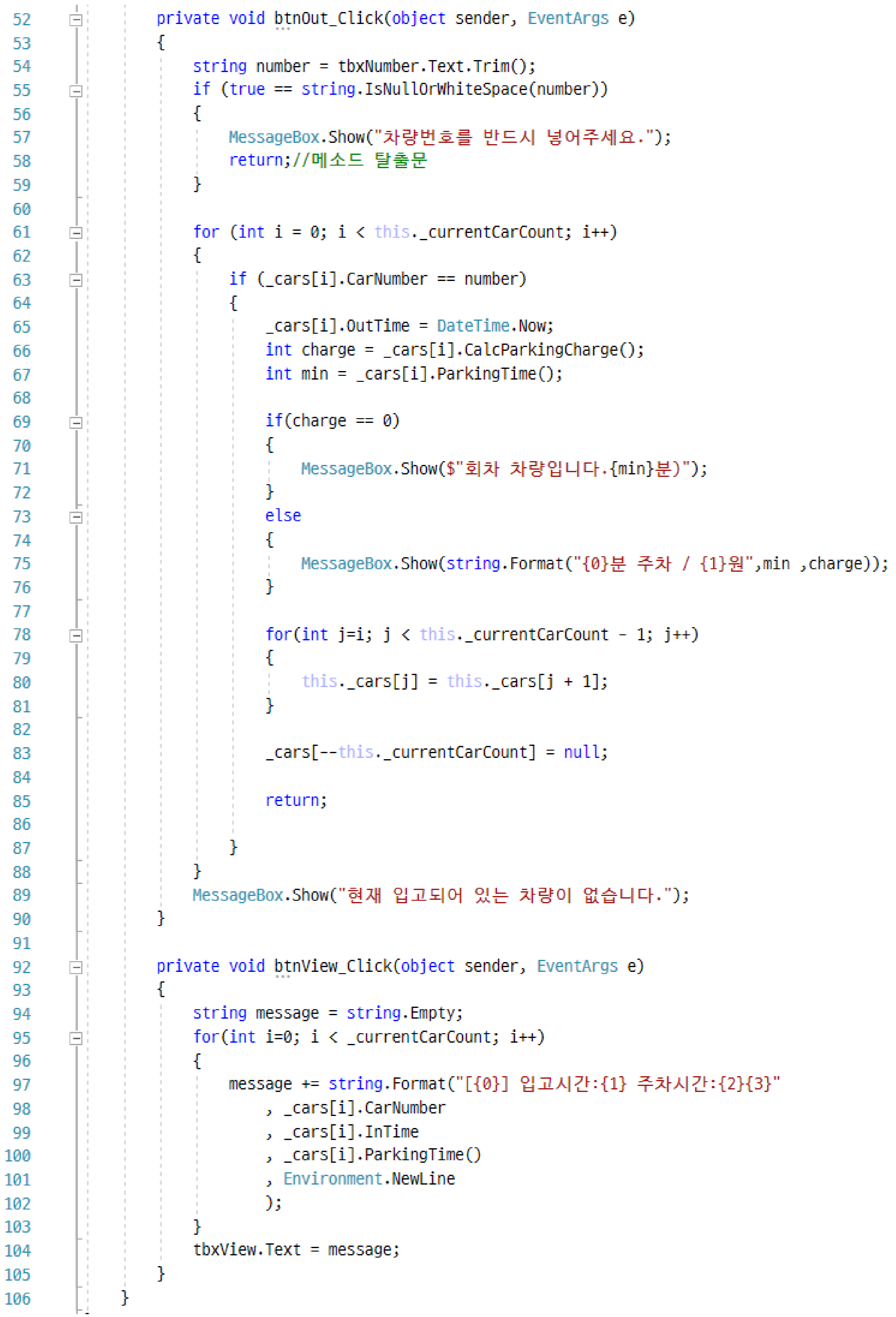


[Week6_1]

Form1.cs

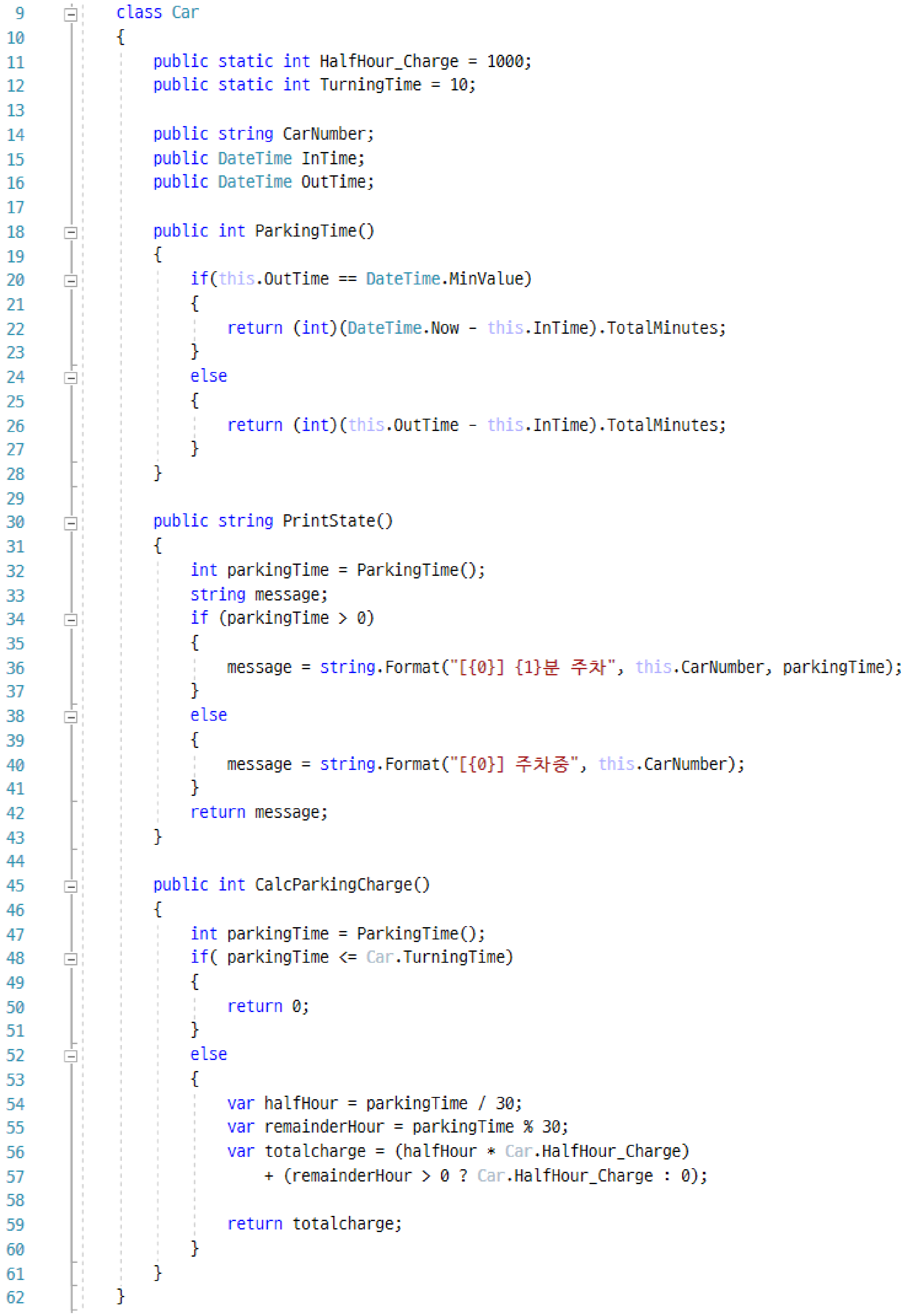


```
    public partial class Form1 : Form
    {
        private Car[] _cars = new Car[10];
        private int _currentCarCount = 0;

        public Form1()
        {
            InitializeComponent();
        }

        private void btnIn_Click(object sender, EventArgs e)
        {
            string number = tbxNumber.Text.Trim();
            if (true == string.IsNullOrWhiteSpace(number))
            {
                MessageBox.Show("차량번호를 반드시 넣어주세요.");
                return;
            }

            for(int i=0; i < this._currentCarCount ; i++)
            {
                if(_cars[i].CarNumber == number)
                {
                    MessageBox.Show("현재 입고되어 있는 차량 번호입니다.");
                    return;
                }
            }

            this._cars[this._currentCarCount] = new Car();
            this._cars[this._currentCarCount].CarNumber = number;

            var now_datetime = (DateTime.Now).AddMinutes(-47);

            this._cars[this._currentCarCount].InTime = now_datetime;
            _currentCarCount++;

            MessageBox.Show($"{number} 차량 입고완료");
        }

```



```
        private void btnOut_Click(object sender, EventArgs e)
        {
            string number = tbxNumber.Text.Trim();
            if (true == string.IsNullOrWhiteSpace(number))
            {
                MessageBox.Show("차량번호를 반드시 넣어주세요.");
                return;//메소드 탈출문
            }

            for (int i = 0; i < this._currentCarCount; i++)
            {
                if (_cars[i].CarNumber == number)
                {
                    _cars[i].OutTime = DateTime.Now;
                    int charge = _cars[i].CalcParkingCharge();
                    int min = _cars[i].ParkingTime();

                    if(charge == 0)
                    {
                        MessageBox.Show($"회차 차량입니다.{min}분)");
                    }
                    else
                    {
                        MessageBox.Show(string.Format("{0}분 주차 / {1}원",min ,charge));
                    }

                    for(int j=i; j < this._currentCarCount - 1; j++)
                    {
                        this._cars[j] = this._cars[j + 1];
                    }

                    _cars[--this._currentCarCount] = null;

                    return;

                }
            }
            MessageBox.Show("현재 입고되어 있는 차량이 없습니다.");
        }

        private void btnView_Click(object sender, EventArgs e)
        {
            string message = string.Empty;
            for(int i=0; i < _currentCarCount; i++)
            {
                message += string.Format("[{0}] 입고시간:{1} 주차시간:{2}{3}"
                    , _cars[i].CarNumber
                    , _cars[i].InTime
                    , _cars[i].ParkingTime()
                    , Environment.NewLine
                    );
            }
            tbxView.Text = message;
        }
    }
```

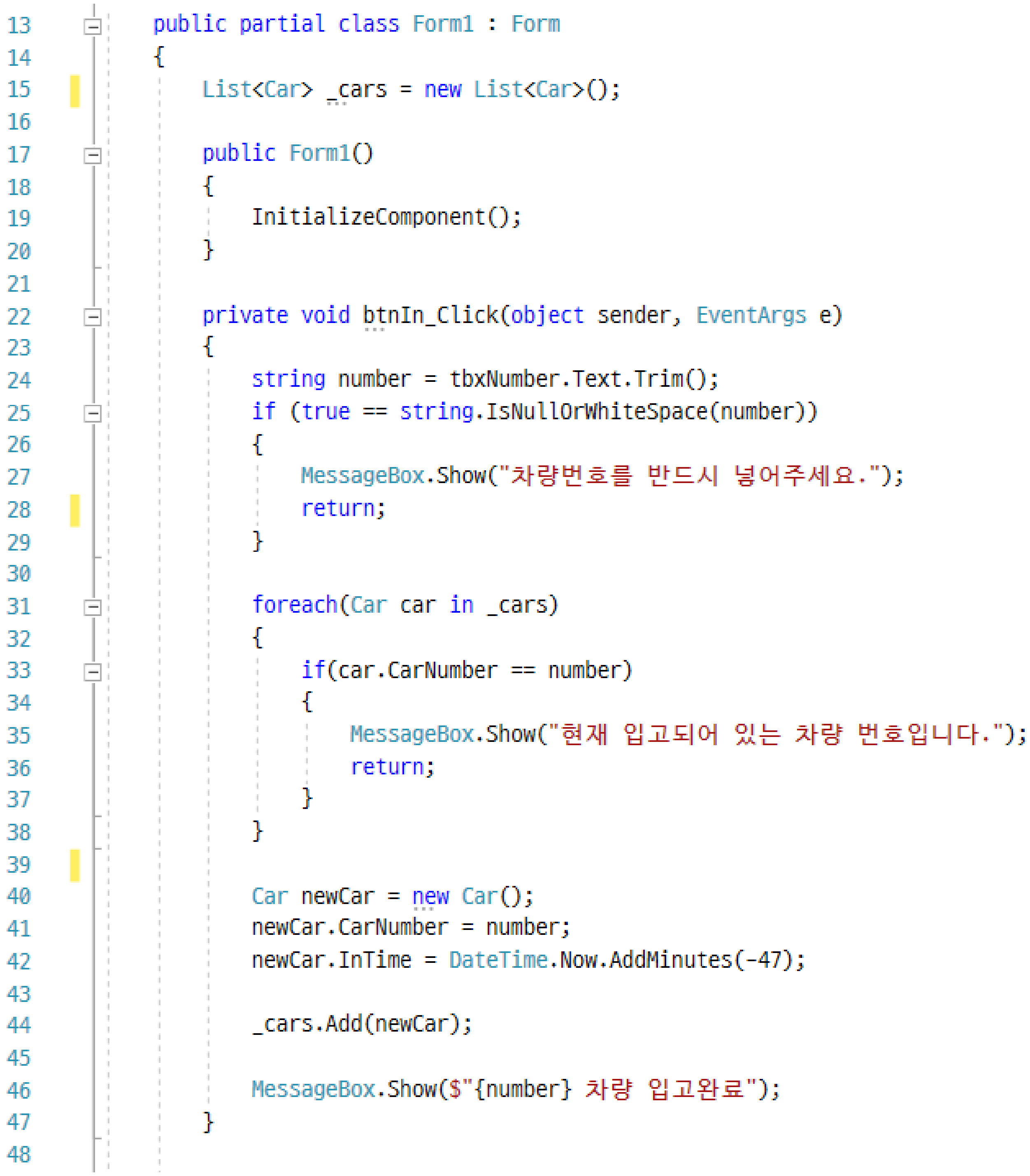
Car.cs



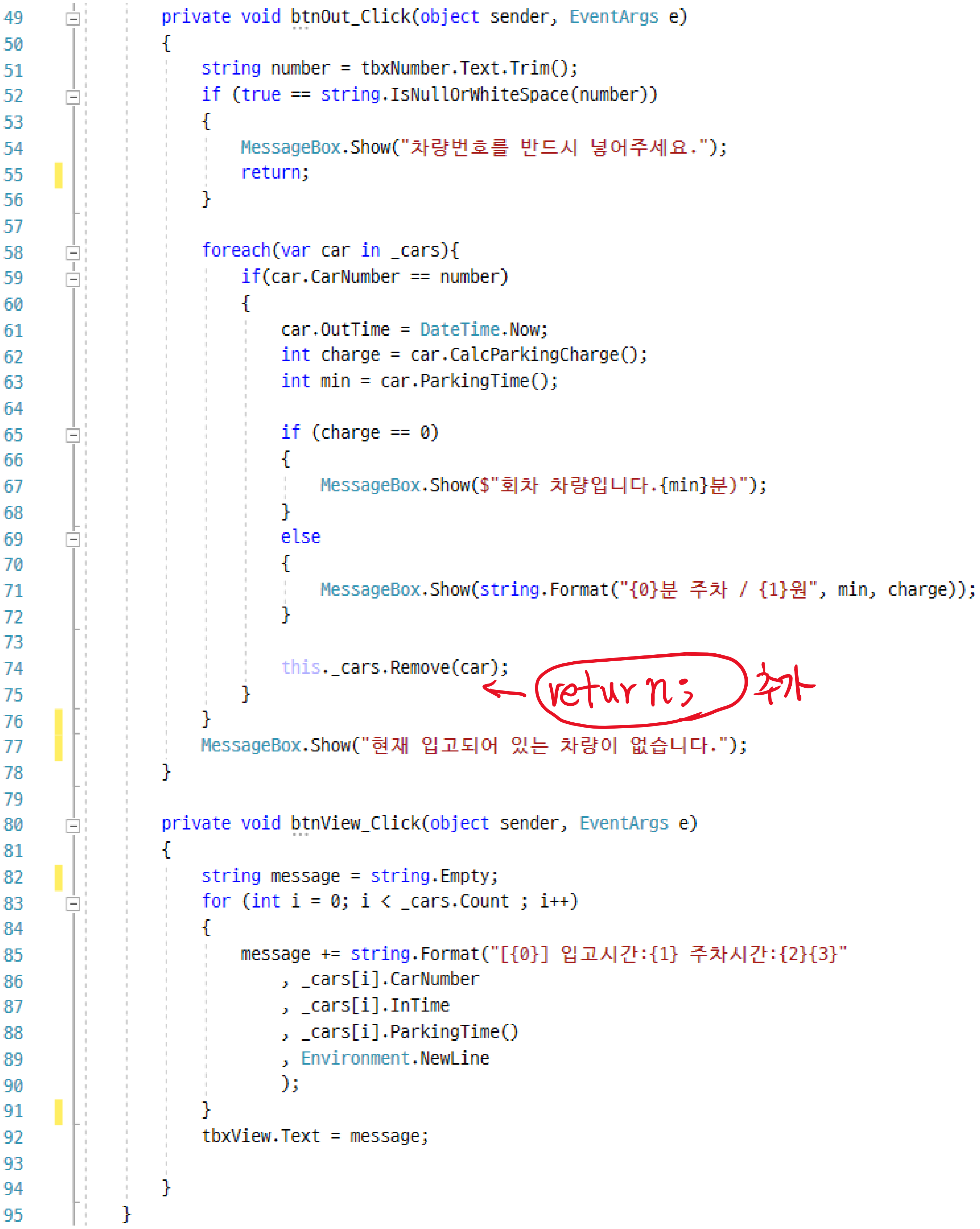
```csharp
class Car
{
    public static int HalfHour_Charge = 1000;
    public static int TurningTime = 10;

    public string CarNumber;
    public DateTime InTime;
    public DateTime OutTime;

    public int ParkingTime()
    {
        if(this.OutTime == DateTime.MinValue)
        {
            return (int)(DateTime.Now - this.InTime).TotalMinutes;
        }
        else
        {
            return (int)(this.OutTime - this.InTime).TotalMinutes;
        }
    }

    public string PrintState()
    {
        int parkingTime = ParkingTime();
        string message;
        if (parkingTime > 0)
        {
            message = string.Format("[{0}] {1}분 주차", this.CarNumber, parkingTime);
        }
        else
        {
            message = string.Format("[{0}] 주차중", this.CarNumber);
        }
        return message;
    }

    public int CalcParkingCharge()
    {
        int parkingTime = ParkingTime();
        if( parkingTime <= Car.TurningTime)
        {
            return 0;
        }
        else
        {
            var halfHour = parkingTime / 30;
            var remainderHour = parkingTime % 30;
            var totalcharge = (halfHour * Car.HalfHour_Charge)
                + (remainderHour > 0 ? Car.HalfHour_Charge : 0);

            return totalcharge;
        }
    }
}
```

[Week6_2]

Form1.cs

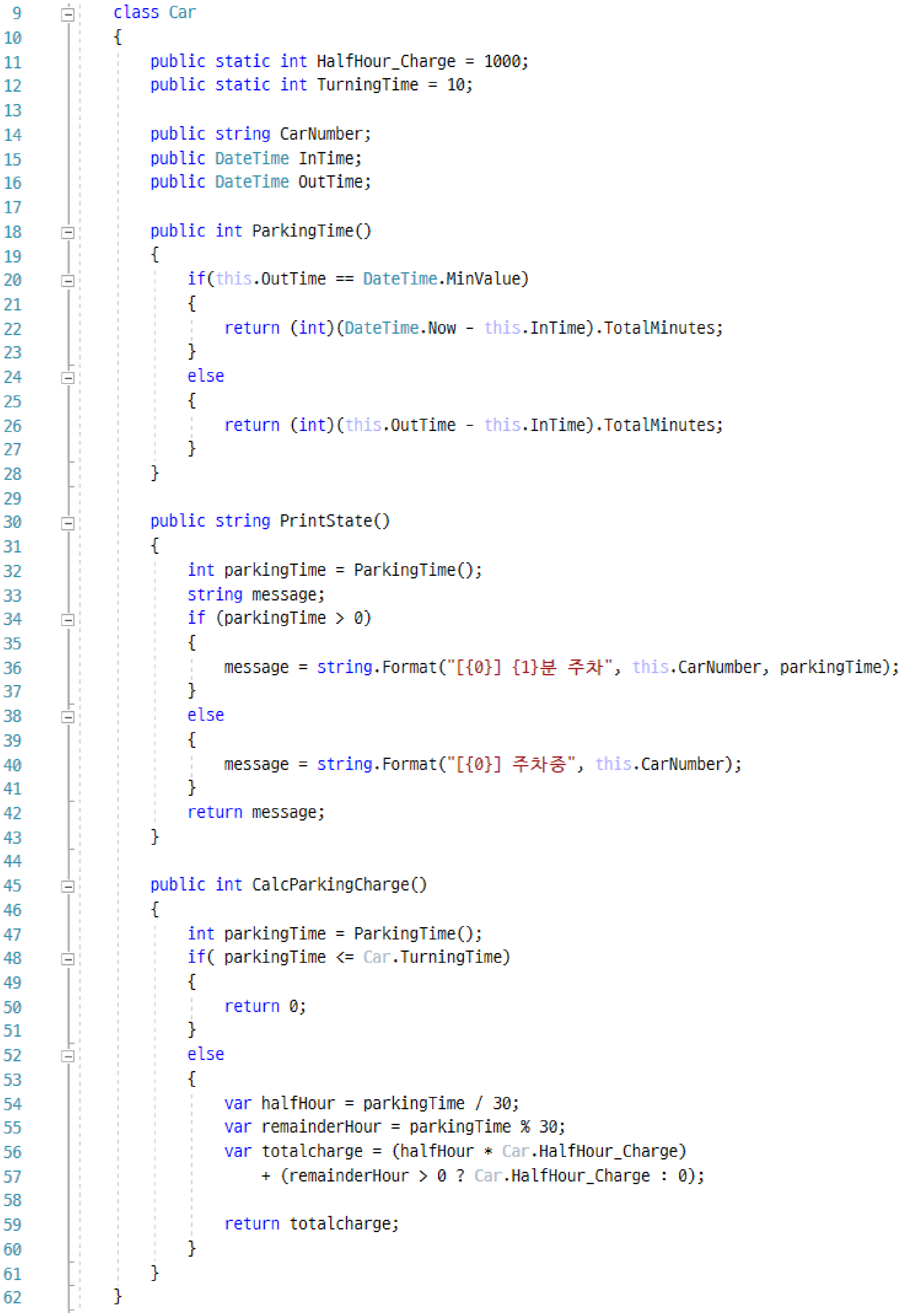


```
public partial class Form1 : Form
{
    List<Car> _cars = new List<Car>();

    public Form1()
    {
        InitializeComponent();
    }

    private void btnIn_Click(object sender, EventArgs e)
    {
        string number = tbxNumber.Text.Trim();
        if (true == string.IsNullOrWhiteSpace(number))
        {
            MessageBox.Show("차량번호를 반드시 넣어주세요.");
            return;
        }

        foreach(Car car in _cars)
        {
            if(car.CarNumber == number)
            {
                MessageBox.Show("현재 입고되어 있는 차량 번호입니다.");
                return;
            }
        }

        Car newCar = new Car();
        newCar.CarNumber = number;
        newCar.InTime = DateTime.Now.AddMinutes(-47);

        _cars.Add(newCar);

        MessageBox.Show($"{number} 차량 입고완료");
    }

```



```
        private void btnOut_Click(object sender, EventArgs e)
        {
            string number = tbxNumber.Text.Trim();
            if (true == string.IsNullOrWhiteSpace(number))
            {
                MessageBox.Show("차량번호를 반드시 넣어주세요.");
                return;
            }

            foreach(var car in _cars){
                if(car.CarNumber == number)
                {
                    car.OutTime = DateTime.Now;
                    int charge = car.CalcParkingCharge();
                    int min = car.ParkingTime();

                    if (charge == 0)
                    {
                        MessageBox.Show($"회차 차량입니다.{min}분)");
                    }
                    else
                    {
                        MessageBox.Show(string.Format("{0}분 주차 / {1}원", min, charge));
                    }

                    this._cars.Remove(car);  ← return; 추가
                }
            }
            MessageBox.Show("현재 입고되어 있는 차량이 없습니다.");
        }

        private void btnView_Click(object sender, EventArgs e)
        {
            string message = string.Empty;
            for (int i = 0; i < _cars.Count ; i++)
            {
                message += string.Format("[{0}] 입고시간:{1} 주차시간:{2}{3}"
                    , _cars[i].CarNumber
                    , _cars[i].InTime
                    , _cars[i].ParkingTime()
                    , Environment.NewLine
                    );
            }
            tbxView.Text = message;

        }
    }
```

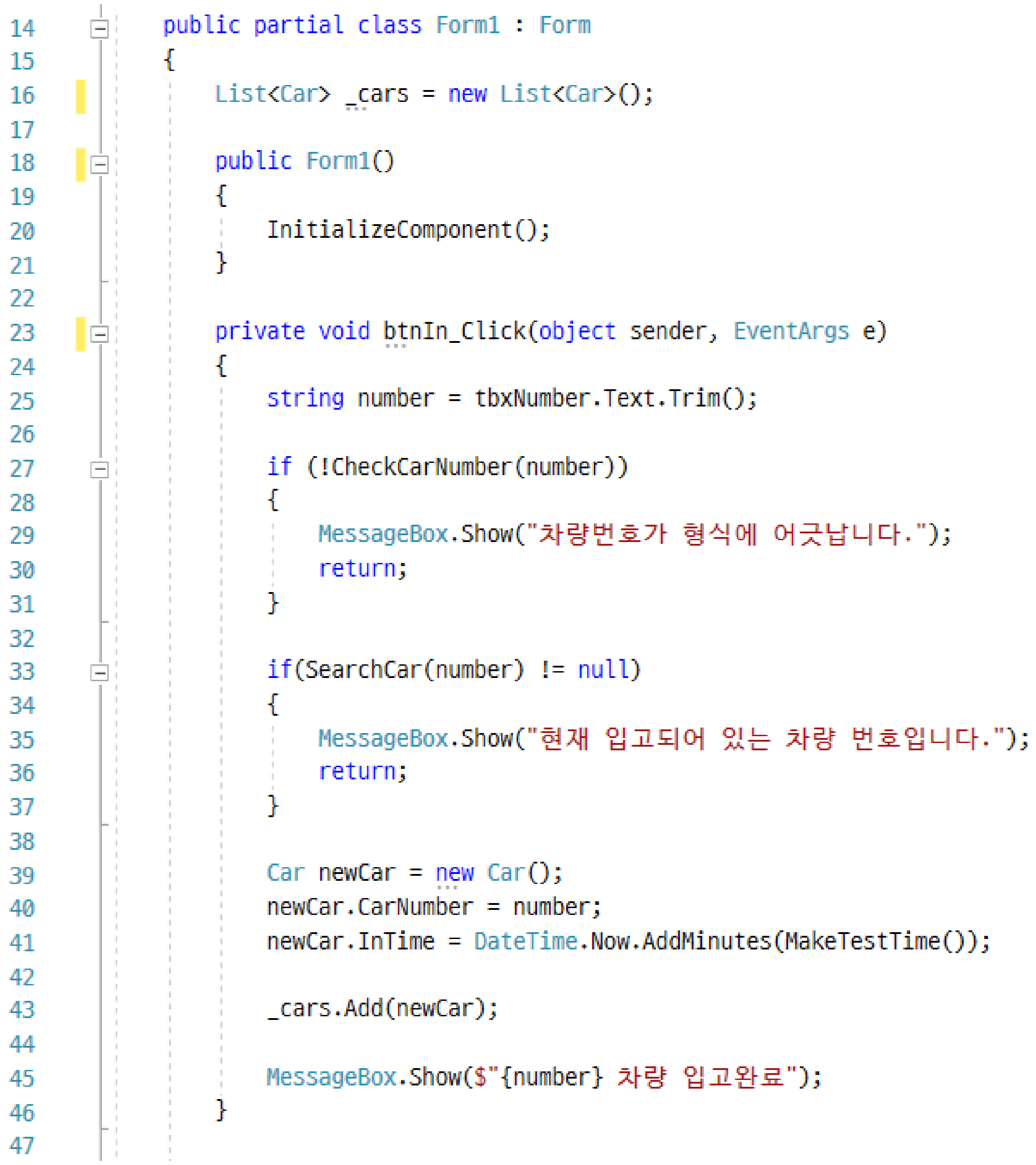
Car.cs – Week6_1과 동일



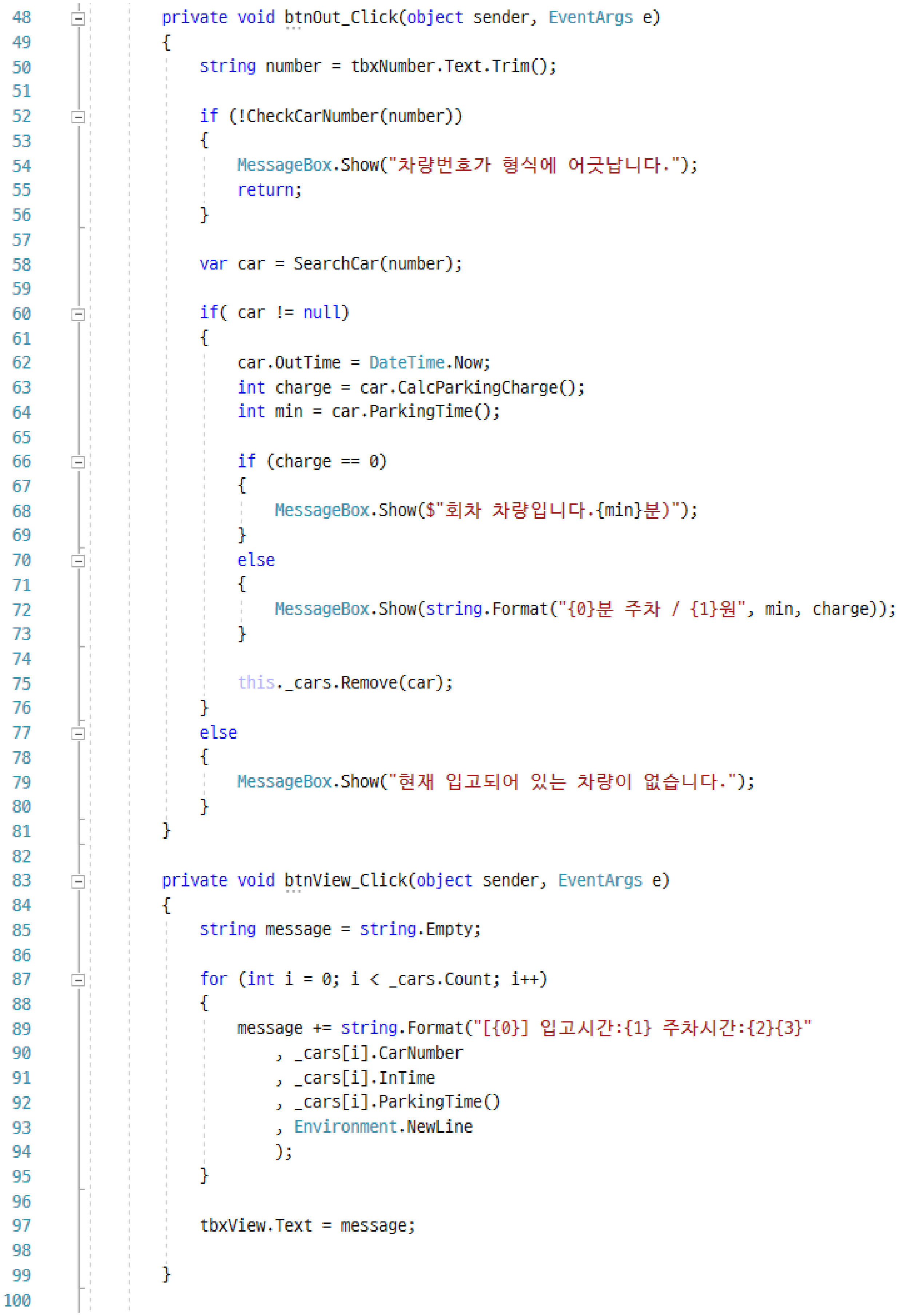
```
class Car
{
    public static int HalfHour_Charge = 1000;
    public static int TurningTime = 10;

    public string CarNumber;
    public DateTime InTime;
    public DateTime OutTime;

    public int ParkingTime()
    {
        if(this.OutTime == DateTime.MinValue)
        {
            return (int)(DateTime.Now - this.InTime).TotalMinutes;
        }
        else
        {
            return (int)(this.OutTime - this.InTime).TotalMinutes;
        }
    }

    public string PrintState()
    {
        int parkingTime = ParkingTime();
        string message;
        if (parkingTime > 0)
        {
            message = string.Format("[{0}] {1}분 주차", this.CarNumber, parkingTime);
        }
        else
        {
            message = string.Format("[{0}] 주차중", this.CarNumber);
        }
        return message;
    }

    public int CalcParkingCharge()
    {
        int parkingTime = ParkingTime();
        if( parkingTime <= Car.TurningTime)
        {
            return 0;
        }
        else
        {
            var halfHour = parkingTime / 30;
            var remainderHour = parkingTime % 30;
            var totalcharge = (halfHour * Car.HalfHour_Charge)
                + (remainderHour > 0 ? Car.HalfHour_Charge : 0);

            return totalcharge;
        }
    }
}
```

[Week6_3]

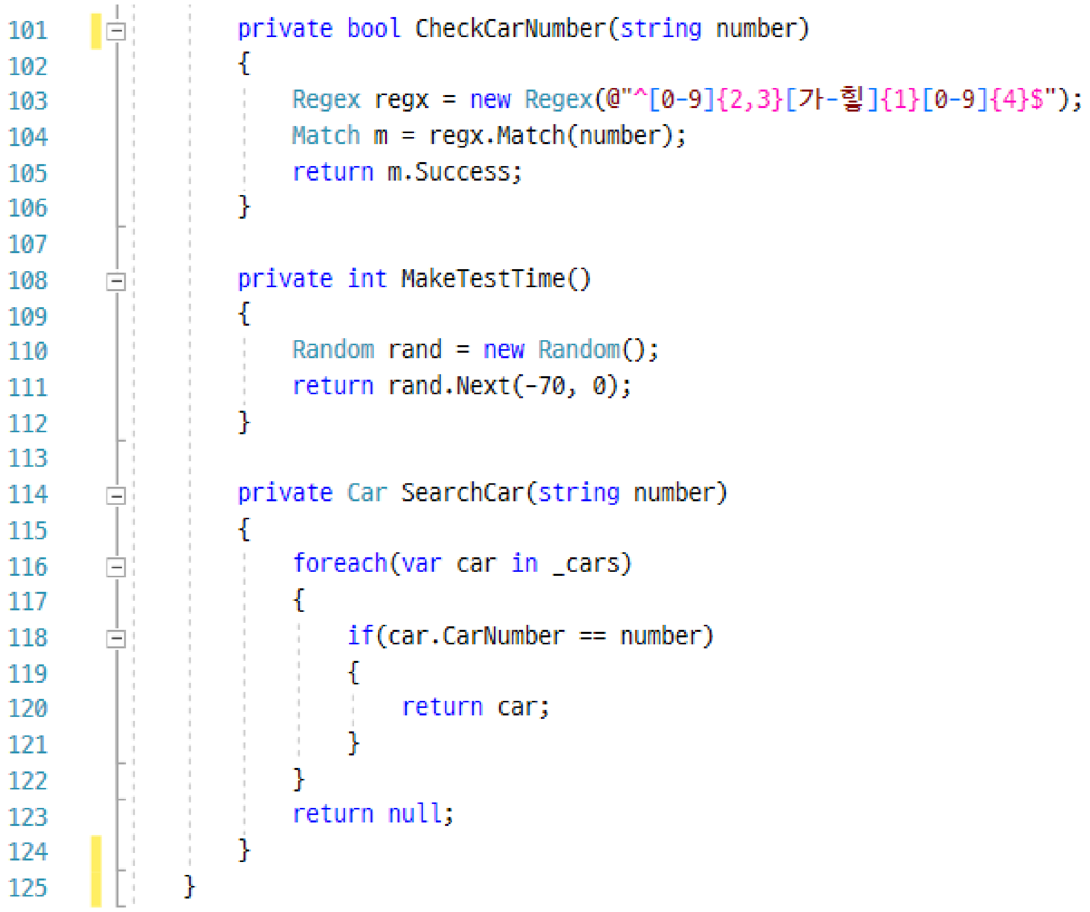
Form1.cs



```
public partial class Form1 : Form
{
    List<Car> _cars = new List<Car>();

    public Form1()
    {
        InitializeComponent();
    }

    private void btnIn_Click(object sender, EventArgs e)
    {
        string number = tbxNumber.Text.Trim();

        if (!CheckCarNumber(number))
        {
            MessageBox.Show("차량번호가 형식에 어긋납니다.");
            return;
        }

        if(SearchCar(number) != null)
        {
            MessageBox.Show("현재 입고되어 있는 차량 번호입니다.");
            return;
        }

        Car newCar = new Car();
        newCar.CarNumber = number;
        newCar.InTime = DateTime.Now.AddMinutes(MakeTestTime());

        _cars.Add(newCar);

        MessageBox.Show($"{number} 차량 입고완료");
    }

```

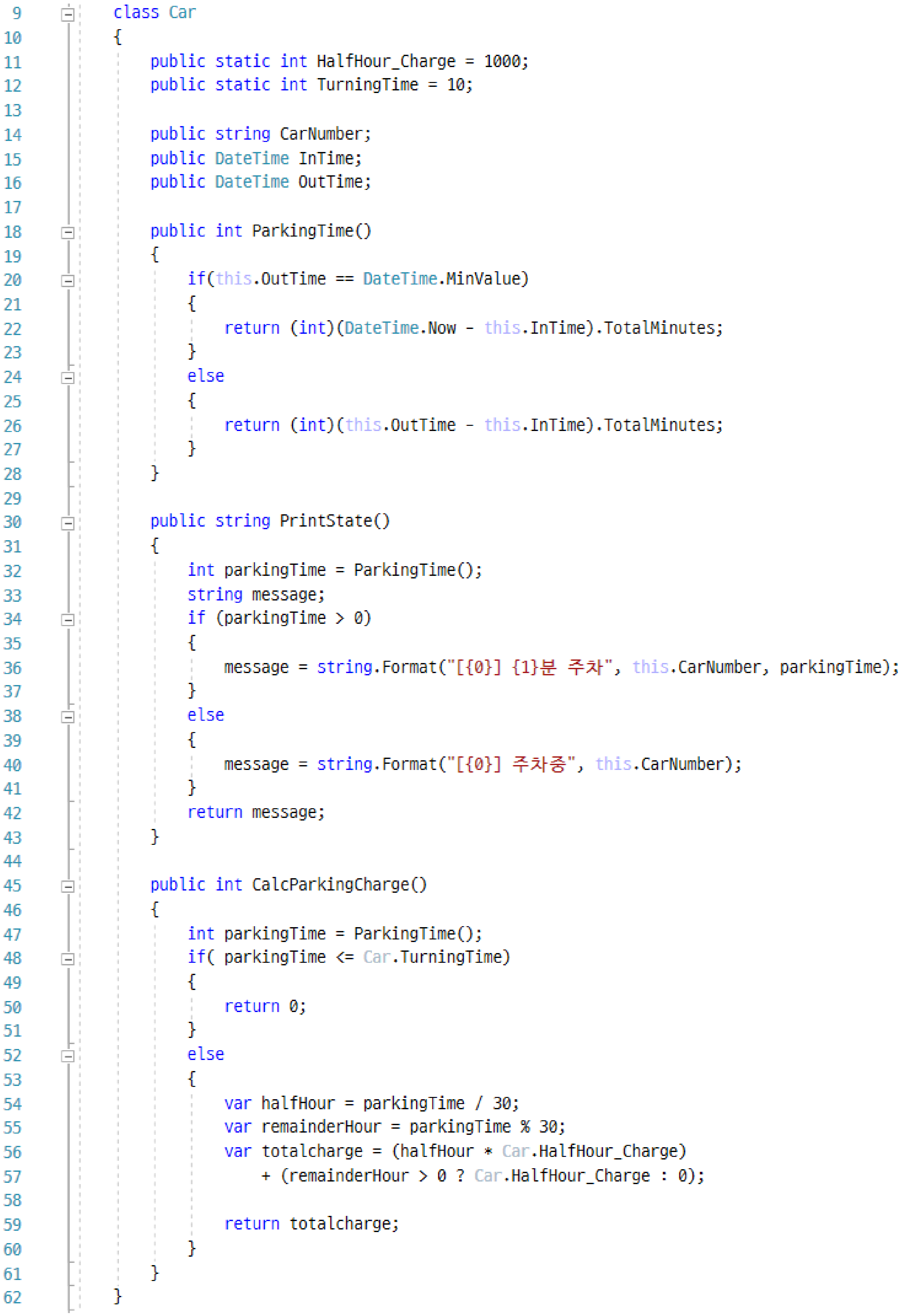


```
        private void btnOut_Click(object sender, EventArgs e)
        {
            string number = tbxNumber.Text.Trim();

            if (!CheckCarNumber(number))
            {
                MessageBox.Show("차량번호가 형식에 어긋납니다.");
                return;
            }

            var car = SearchCar(number);

            if( car != null)
            {
                car.OutTime = DateTime.Now;
                int charge = car.CalcParkingCharge();
                int min = car.ParkingTime();

                if (charge == 0)
                {
                    MessageBox.Show($"회차 차량입니다.{min}분)");
                }
                else
                {
                    MessageBox.Show(string.Format("{0}분 주차 / {1}원", min, charge));
                }

                this._cars.Remove(car);
            }
            else
            {
                MessageBox.Show("현재 입고되어 있는 차량이 없습니다.");
            }
        }

        private void btnView_Click(object sender, EventArgs e)
        {
            string message = string.Empty;

            for (int i = 0; i < _cars.Count; i++)
            {
                message += string.Format("[{0}] 입고시간:{1} 주차시간:{2}{3}"
                    , _cars[i].CarNumber
                    , _cars[i].InTime
                    , _cars[i].ParkingTime()
                    , Environment.NewLine
                    );
            }

            tbxView.Text = message;

        }

```



```
        private bool CheckCarNumber(string number)
        {
            Regex regx = new Regex(@"^[0-9]{2,3}[가-힣]{1}[0-9]{4}$");
            Match m = regx.Match(number);
            return m.Success;
        }

        private int MakeTestTime()
        {
            Random rand = new Random();
            return rand.Next(-70, 0);
        }

        private Car SearchCar(string number)
        {
            foreach(var car in _cars)
            {
                if(car.CarNumber == number)
                {
                    return car;
                }
            }
            return null;
        }
    }
```

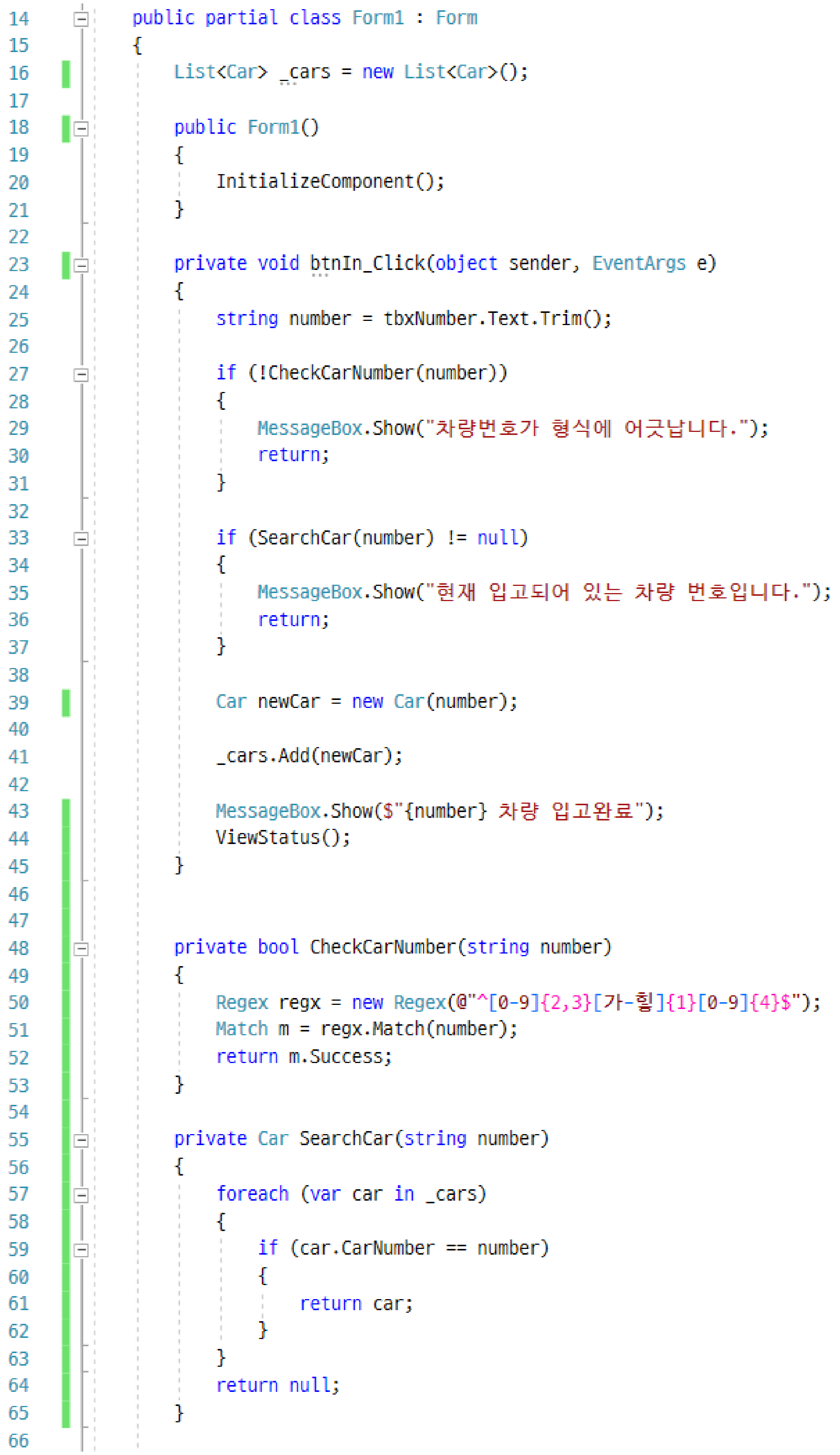
Car.cs – Week6_1과 동일



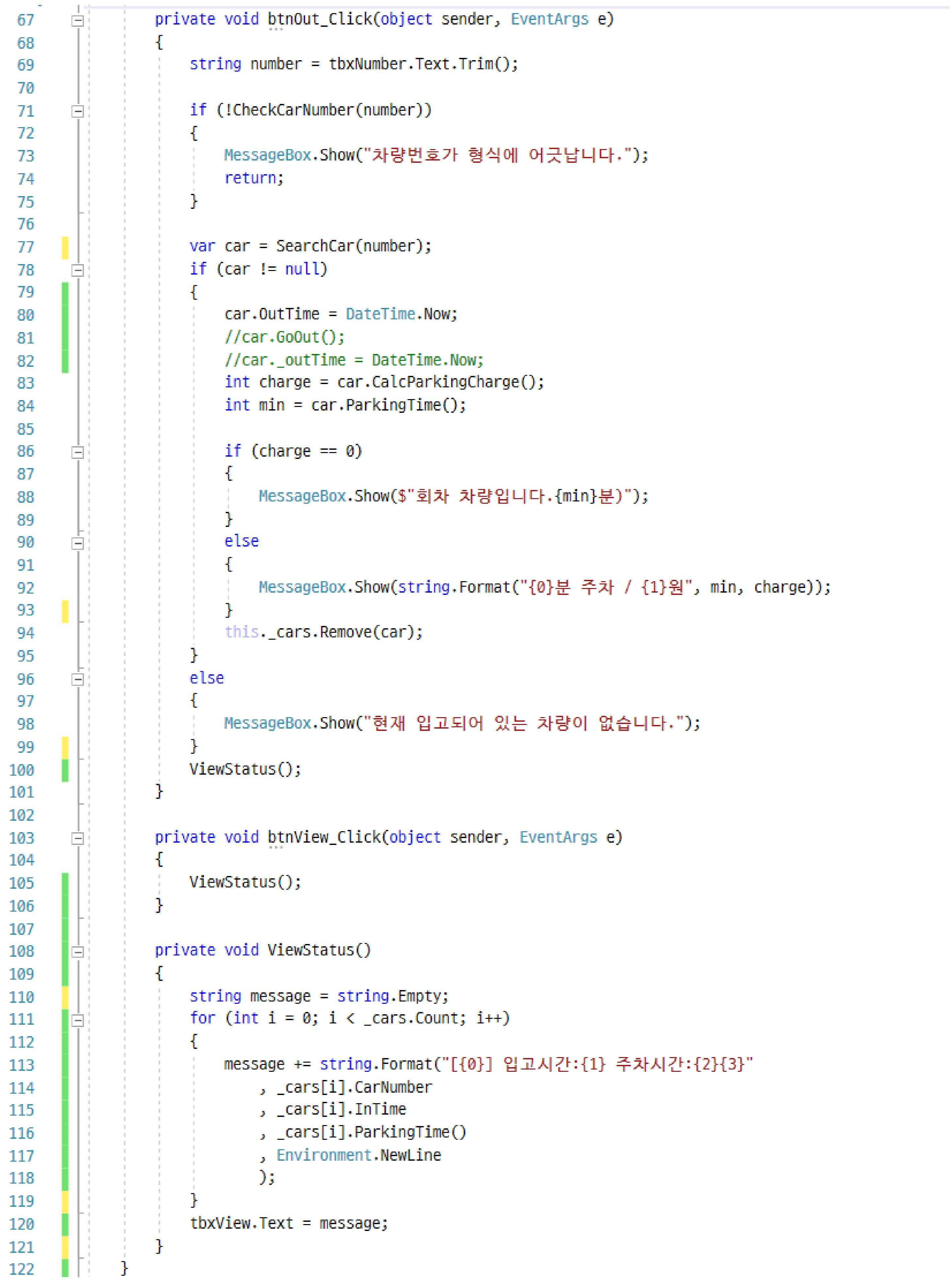
```
class Car
{
    public static int HalfHour_Charge = 1000;
    public static int TurningTime = 10;

    public string CarNumber;
    public DateTime InTime;
    public DateTime OutTime;

    public int ParkingTime()
    {
        if(this.OutTime == DateTime.MinValue)
        {
            return (int)(DateTime.Now - this.InTime).TotalMinutes;
        }
        else
        {
            return (int)(this.OutTime - this.InTime).TotalMinutes;
        }
    }

    public string PrintState()
    {
        int parkingTime = ParkingTime();
        string message;
        if (parkingTime > 0)
        {
            message = string.Format("[{0}] {1}분 주차", this.CarNumber, parkingTime);
        }
        else
        {
            message = string.Format("[{0}] 주차중", this.CarNumber);
        }
        return message;
    }

    public int CalcParkingCharge()
    {
        int parkingTime = ParkingTime();
        if( parkingTime <= Car.TurningTime)
        {
            return 0;
        }
        else
        {
            var halfHour = parkingTime / 30;
            var remainderHour = parkingTime % 30;
            var totalcharge = (halfHour * Car.HalfHour_Charge)
                + (remainderHour > 0 ? Car.HalfHour_Charge : 0);

            return totalcharge;
        }
    }
}
```

[Week6_4]

Form1.cs – 강의 내용과 메소드 순서가 좀 다름 (순서 상관 없음)

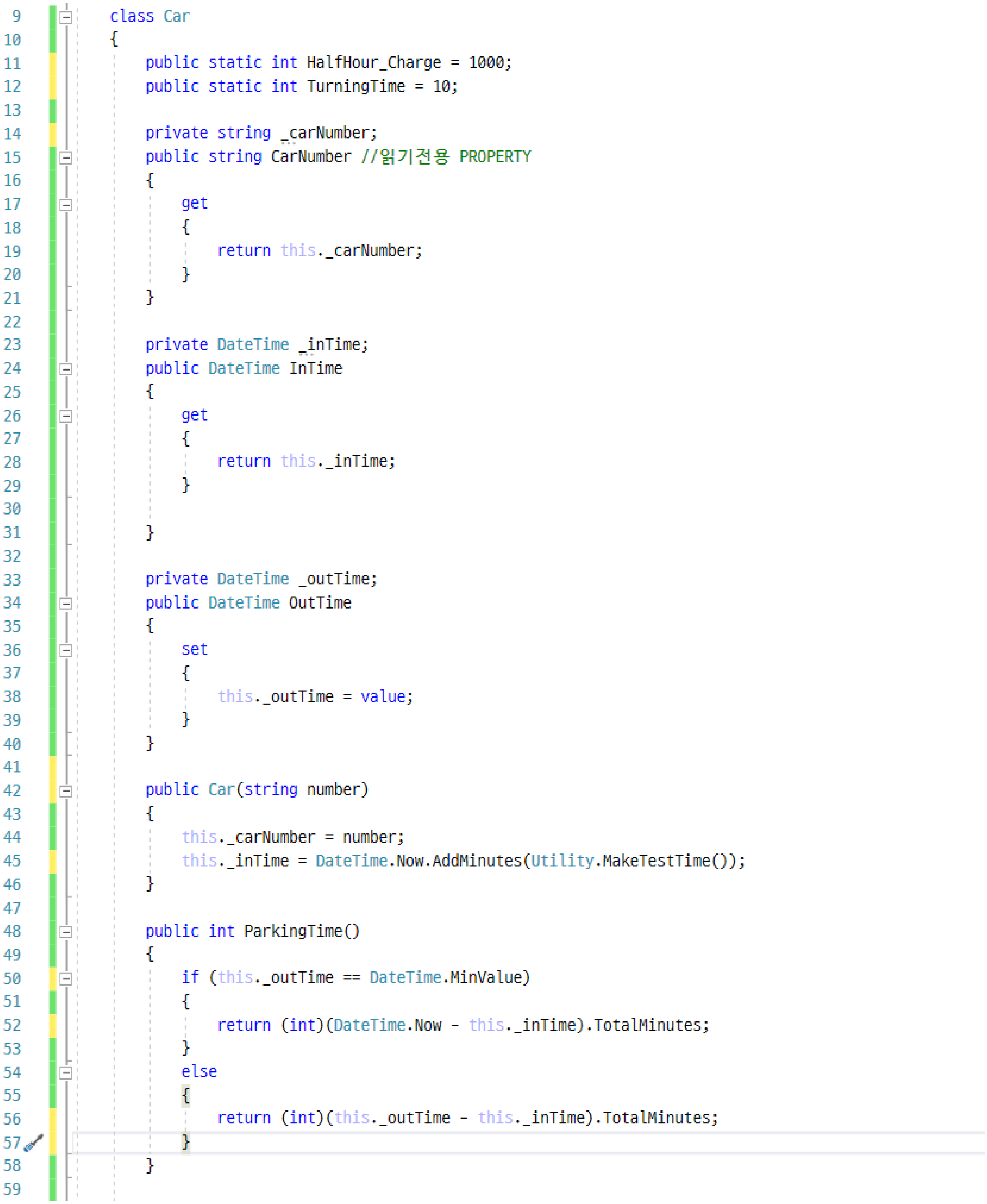


```
    public partial class Form1 : Form
    {
        List<Car> _cars = new List<Car>();

        public Form1()
        {
            InitializeComponent();
        }

        private void btnIn_Click(object sender, EventArgs e)
        {
            string number = tbxNumber.Text.Trim();

            if (!CheckCarNumber(number))
            {
                MessageBox.Show("차량번호가 형식에 어긋납니다.");
                return;
            }

            if (SearchCar(number) != null)
            {
                MessageBox.Show("현재 입고되어 있는 차량 번호입니다.");
                return;
            }

            Car newCar = new Car(number);

            _cars.Add(newCar);

            MessageBox.Show($"{number} 차량 입고완료");
            ViewStatus();
        }


        private bool CheckCarNumber(string number)
        {
            Regex regx = new Regex(@"^[0-9]{2,3}[가-힣]{1}[0-9]{4}$");
            Match m = regx.Match(number);
            return m.Success;
        }

        private Car SearchCar(string number)
        {
            foreach (var car in _cars)
            {
                if (car.CarNumber == number)
                {
                    return car;
                }
            }
            return null;
        }

```

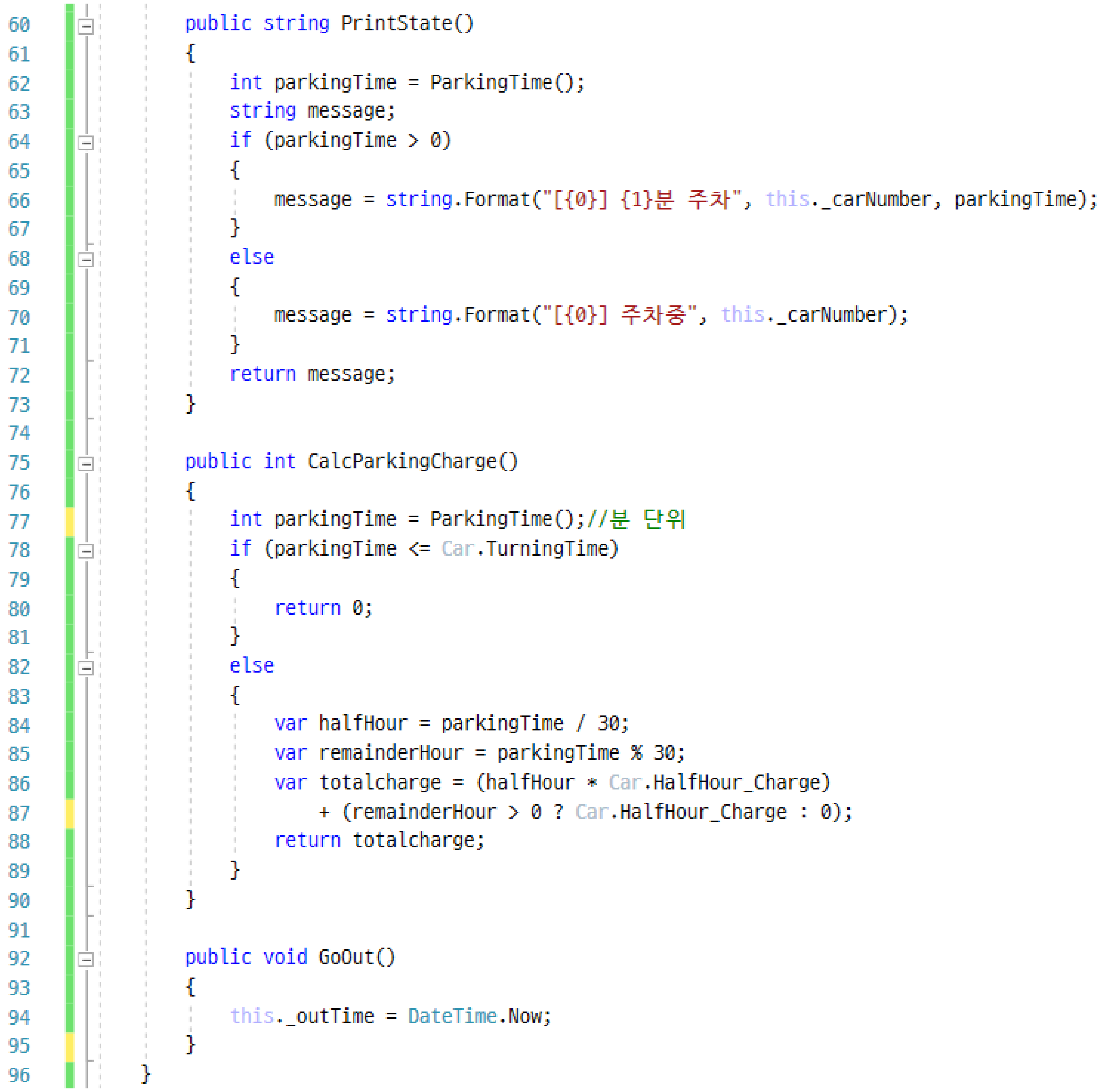


```csharp
        private void btnOut_Click(object sender, EventArgs e)
        {
            string number = tbxNumber.Text.Trim();

            if (!CheckCarNumber(number))
            {
                MessageBox.Show("차량번호가 형식에 어긋납니다.");
                return;
            }

            var car = SearchCar(number);
            if (car != null)
            {
                car.OutTime = DateTime.Now;
                //car.GoOut();
                //car._outTime = DateTime.Now;
                int charge = car.CalcParkingCharge();
                int min = car.ParkingTime();

                if (charge == 0)
                {
                    MessageBox.Show($"회차 차량입니다.{min}분)");
                }
                else
                {
                    MessageBox.Show(string.Format("{0}분 주차 / {1}원", min, charge));
                }
                this._cars.Remove(car);
            }
            else
            {
                MessageBox.Show("현재 입고되어 있는 차량이 없습니다.");
            }
            ViewStatus();
        }

        private void btnView_Click(object sender, EventArgs e)
        {
            ViewStatus();
        }

        private void ViewStatus()
        {
            string message = string.Empty;
            for (int i = 0; i < _cars.Count; i++)
            {
                message += string.Format("[{0}] 입고시간:{1} 주차시간:{2}{3}"
                    , _cars[i].CarNumber
                    , _cars[i].InTime
                    , _cars[i].ParkingTime()
                    , Environment.NewLine
                    );
            }
            tbxView.Text = message;
        }
    }
```

Car.cs

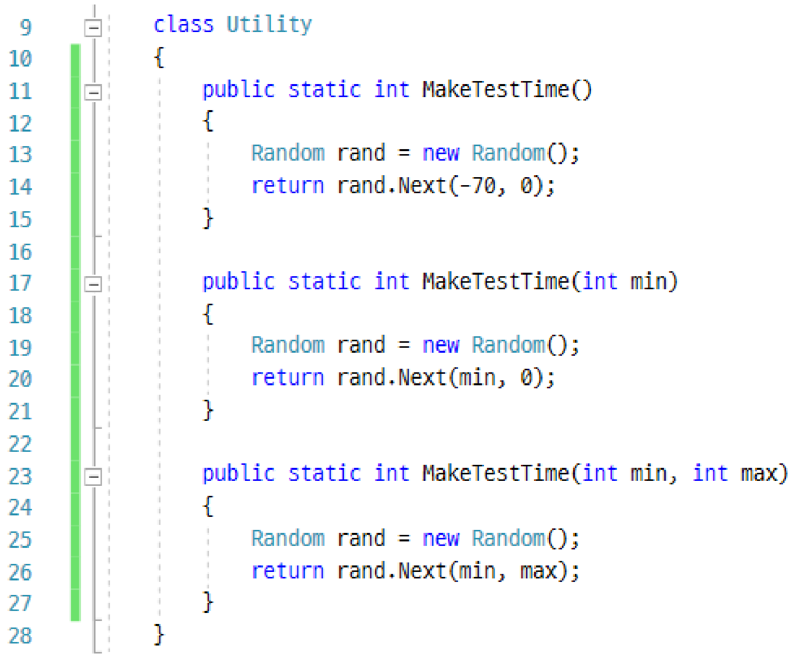


```
class Car
{
    public static int HalfHour_Charge = 1000;
    public static int TurningTime = 10;

    private string _carNumber;
    public string CarNumber //읽기전용 PROPERTY
    {
        get
        {
            return this._carNumber;
        }
    }

    private DateTime _inTime;
    public DateTime InTime
    {
        get
        {
            return this._inTime;
        }

    }

    private DateTime _outTime;
    public DateTime OutTime
    {
        set
        {
            this._outTime = value;
        }
    }

    public Car(string number)
    {
        this._carNumber = number;
        this._inTime = DateTime.Now.AddMinutes(Utility.MakeTestTime());
    }

    public int ParkingTime()
    {
        if (this._outTime == DateTime.MinValue)
        {
            return (int)(DateTime.Now - this._inTime).TotalMinutes;
        }
        else
        {
            return (int)(this._outTime - this._inTime).TotalMinutes;
        }
    }

```



```
        public string PrintState()
        {
            int parkingTime = ParkingTime();
            string message;
            if (parkingTime > 0)
            {
                message = string.Format("[{0}] {1}분 주차", this._carNumber, parkingTime);
            }
            else
            {
                message = string.Format("[{0}] 주차중", this._carNumber);
            }
            return message;
        }

        public int CalcParkingCharge()
        {
            int parkingTime = ParkingTime();//분 단위
            if (parkingTime <= Car.TurningTime)
            {
                return 0;
            }
            else
            {
                var halfHour = parkingTime / 30;
                var remainderHour = parkingTime % 30;
                var totalcharge = (halfHour * Car.HalfHour_Charge)
                    + (remainderHour > 0 ? Car.HalfHour_Charge : 0);
                return totalcharge;
            }
        }

        public void GoOut()
        {
            this._outTime = DateTime.Now;
        }
    }
```

Utility.cs



```
    class Utility
    {
        public static int MakeTestTime()
        {
            Random rand = new Random();
            return rand.Next(-70, 0);
        }

        public static int MakeTestTime(int min)
        {
            Random rand = new Random();
            return rand.Next(min, 0);
        }

        public static int MakeTestTime(int min, int max)
        {
            Random rand = new Random();
            return rand.Next(min, max);
        }
    }
```