< Summary

Information

Class:	CarNS.Car
Assembly:	Car
File(s):	D:\Users\Null.HOME-PC\Documents\GitHub\QA\lab6\Car\Car.cs

Line coverage

96%

Covered lines:	83
Uncovered lines:	3
Coverable lines:	86
Total lines:	161
Line coverage:	96.5%

Branch coverage

95%

Covered branches:	42
Total branches:	44
Branch coverage:	95.4%

Method coverage

Methods/Properties

- .ctor()
- TurnOnEngine()
- TurnOffEngine()
- CheckSpeedRange(System.Int32,Syste…
- SetGear(System.Int32)
- SetSpeed(System.Int32)
- IsTurnedOn()
- GetDirection()
- GetSpeed()
- GetGear()
- Main(System.String[])

Metrics

Method	Branch coverage	Crap Score	Cyclomatic complexity	Line covera
.ctor()	100%	1	1	100%
TurnOnEngine()	100%	1	1	100%
TurnOffEngine()	100%	4	4	100%
CheckSpeedRange(...)	100%	2	2	100%
SetGear(...)	100%	16	16	100%
SetSpeed(...)	88.88%	18.03	18	95.45%
IsTurnedOn()	100%	1	1	100%
GetDirection()	100%	4	4	100%
GetSpeed()	100%	1	1	100%
GetGear()	100%	1	1	100%
Main(...)	100%	2	1	0%

File(s)

D:\Users\Null.HOME-PC\Documents\GitHub\QA\lab6\Car\Car.cs

#	Line	Line coverage
	1	`namespace CarNS`
	2	`{`
	3	`    public enum Direction`
	4	`    {`
	5	`        FORWARD,`

Methods/Properties

- .ctor()
- TurnOnEngine()
- TurnOffEngine()
- CheckSpeedRange(System.Int32,Syste…
- SetGear(System.Int32)
- SetSpeed(System.Int32)
- IsTurnedOn()
- GetDirection()
- GetSpeed()
- GetGear()
- Main(System.String[])

#	Line	Line coverage
	6	BACKWARD,
	7	IMMOBILE
	8	}
	9	
	10	public class Car
	11	{
24	12	private int m_gear = 0;
24	13	private int m_speed = 0;
24	14	private bool m_isEngineOn = false;
	15	
24	16	private readonly Dictionary<int, Tuple<int, int>> gearSpeedRange = new I
24	17	{
24	18	{ -1, Tuple.Create(-20, 0) },
24	19	{ 0, Tuple.Create(0, 150) },
24	20	{ 1, Tuple.Create(0, 30) },
24	21	{ 2, Tuple.Create(20, 50) },
24	22	{ 3, Tuple.Create(30, 60) },
24	23	{ 4, Tuple.Create(40, 90) },
24	24	{ 5, Tuple.Create(50, 150) }
24	25	};
	26	
	27	public bool TurnOnEngine()
22	28	{
22	29	m_isEngineOn = true;
22	30	return true;
22	31	}
	32	
	33	public bool TurnOffEngine()
3	34	{
3	35	if ((m_gear != 0) || (m_speed != 0))
2	36	{
2	37	return false;
	38	}
	39	
1	40	m_isEngineOn = false;

Methods/Properties

- .ctor()
- TurnOnEngine()
- TurnOffEngine()
- CheckSpeedRange(System.Int32,Syste…
- SetGear(System.Int32)
- SetSpeed(System.Int32)
- IsTurnedOn()
- GetDirection()
- GetSpeed()
- GetGear()
- Main(System.String[])

#	Line	Line coverage
1	41	return true;
3	42	}
	43	
	44	private bool CheckSpeedRange(int speed, int gear)
11	45	{
11	46	var range = gearSpeedRange[gear];
11	47	return speed >= range.Item1 && speed <= range.Item2;
11	48	}
	49	
	50	public bool SetGear(int gear)
20	51	{
20	52	if (!IsTurnedOn())
1	53	{
1	54	return false;
	55	}
	56	
19	57	switch (gear)
	58	{
	59	case -1:
8	60	if (m_speed != 0 && m_gear != gear)
1	61	{
1	62	return false;
	63	}
7	64	break;
	65	
	66	case 0:
1	67	break;
	68	
	69	case 1:
	70	case 2:
	71	case 3:
	72	case 4:
	73	case 5:
8	74	if (!CheckSpeedRange(m_speed, gear))
2	75	{

Methods/Properties

- .ctor()
- TurnOnEngine()
- TurnOffEngine()
- CheckSpeedRange(System.Int32,Syste…
- SetGear(System.Int32)
- SetSpeed(System.Int32)
- IsTurnedOn()
- GetDirection()
- GetSpeed()
- GetGear()
- Main(System.String[])

#	Line	Line coverage
2	76	return false;
	77	}
6	78	break;
	79	
	80	default:
2	81	return false;
	82	}
	83	
14	84	m_gear = gear;
14	85	return true;
20	86	}
	87	
	88	public bool SetSpeed(int speed)
13	89	{
13	90	if (speed < 0)
2	91	{
2	92	return false;
	93	}
	94	
11	95	switch (m_gear)
	96	{
	97	case -1:
4	98	if (-speed < gearSpeedRange[-1].Item1)
1	99	{
1	100	return false;
	101	}
3	102	m_speed = -speed;
3	103	return true;
	104	
	105	case 0:
4	106	if (speed > Math.Abs(m_speed))
3	107	{
3	108	return false;
	109	}
1	110	m_speed = (m_speed < 0) ? -speed : speed;

Methods/Properties

- .ctor()
- TurnOnEngine()
- TurnOffEngine()
- CheckSpeedRange(System.Int32,Syste…
- SetGear(System.Int32)
- SetSpeed(System.Int32)
- IsTurnedOn()
- GetDirection()
- GetSpeed()
- GetGear()
- Main(System.String[])

#	Line	Line coverage
1	111	return true;
	112	
	113	case 1:
	114	case 2:
	115	case 3:
	116	case 4:
	117	case 5:
3	118	if (!CheckSpeedRange(speed, m_gear))
1	119	{
1	120	return false;
	121	}
2	122	m_speed = speed;
2	123	return true;
	124	
	125	//сюда мы никогда не попадём, т.к передача не может быть невали,
	126	default:
0	127	return false;
	128	}
13	129	}
	130	
	131	public bool IsTurnedOn()
26	132	{
26	133	return m_isEngineOn;
26	134	}
	135	
	136	public Direction GetDirection()
4	137	{
4	138	if (m_speed > 0)
1	139	return Direction.FORWARD;
3	140	else if (m_speed == 0)
2	141	return Direction.IMMOBILE;
	142	else
1	143	return Direction.BACKWARD;
4	144	}
	145	

Methods/Properties

- .ctor()
- TurnOnEngine()
- TurnOffEngine()
- CheckSpeedRange(System.Int32,Syste…
- SetGear(System.Int32)
- SetSpeed(System.Int32)
- IsTurnedOn()
- GetDirection()
- GetSpeed()
- GetGear()
- Main(System.String[])

#	Line	Line coverage
	146	public int GetSpeed()
8	147	{
8	148	return m_speed;
8	149	}
	150	
	151	public int GetGear()
8	152	{
8	153	return m_gear;
8	154	}
	155	
	156	public static void Main(string[] args)
0	157	{
	158	
0	159	}
	160	}
	161	}

Methods/Properties

- .ctor()
- TurnOnEngine()
- TurnOffEngine()
- CheckSpeedRange(System.Int32,Syste…
- SetGear(System.Int32)
- SetSpeed(System.Int32)
- IsTurnedOn()
- GetDirection()
- GetSpeed()
- GetGear()
- Main(System.String[])

Generated by: ReportGenerator 5.2.5.0
15.05.2024 - 12:33:22
GitHub | reportgenerator.io