



ДЕПАРТАМЕНТ ОБРАЗОВАНИЯ И НАУКИ

ГОРОДА МОСКВЫ

Государственное бюджетное профессиональное
образовательное учреждение города Москвы

«Колледж малого бизнеса № 4»

(ГБПОУ КМБ № 4)

ОСНОВНЫЕ ПОНЯТИЯ ОБЪЕКТНО- ОРИЕНТИРОВАННОГО ПРОГРАММИРОВАНИЯ.

Специальность: 09.02.07 Информационные системы и программирование

Форма обучения: очная

Студент(ка): Ермакова Анастасия Юрьевна

Группа: ИПО-21.24

Руководитель: Рыбаков Александр Сергеевич

Отчётная работа защищена с оценкой «____» _____

Москва, 2025 г.

Оглавление

1.Инкапсуляция	3
2. Конструктор класса.....	5
3.Свойства класса.....	7
4. Наследование.....	9
5.Полиморфизм	11

1.Инкапсуляция

- Код:

```
1   using System;
2
3   // 2.1 Student
4   class Student
5   {
6       public string name;
7   }
8
9   // 2.2 Car
10  class Car
11  {
12      public string year;
13  }
14
15  // 2.3 Point
16  class Point
17  {
18      public string x;
19  }
20
21  // 2.4 Person
22  class Person
23  {
24      public string age;
25
26      Ссылка: 1
27      public void Print()
28      {
29          Console.WriteLine(age);
30      }
31
32  // 2.5 Table
33  class Table
```

```
34  {
35      public string rows;
36      public string cols;
37
38      Ссылка: 1
39      public void Display()
40      {
41          Console.WriteLine(rows);
42          Console.WriteLine(cols);
43      }
44
45  // 2.6 Manager
46  class Manager
47  {
48      public string age;
49      public string name;
50
51      Ссылка: 1
52      public string GetAge()
53      {
54          return age;
55      }
56
57      Ссылка: 1
58      public string GetName()
59      {
60          return name;
61      }
62  // 2.7 Point3D
```

```
63     v class Point3D
64     {
65         public string x;
66         public string y;
67         public string z;
68
69         v Ссылок: 1
70         public void Show()
71         {
72             Console.WriteLine(x);
73             Console.WriteLine(y);
74             Console.WriteLine(z);
75         }
76
77         // 2.8 Shop
78         v Ссылок: 2
79         class Shop
80         {
81             public string name;
82
83             v Ссылок: 1
84             public string GetName()
85             {
86                 return name;
87             }
88
89             v Ссылок: 1
90             public void SetName(string newName)
91             {
92                 name = newName;
93             }
94
95             Ссылок: 0
```

```
93     v class Program
94     {
95         v Ссылок: 0
96         static void Main()
97         {
98             // Student
99             Student student = new Student();
100            student.name = "TASK_1      Name: Ivan";
101            Console.WriteLine(student.name);
102
103            // Car
104            Car car = new Car();
105            car.year = "TASK_2      Year: 2020";
106            Console.WriteLine(car.year);
107
108            // Point
109            Point point = new Point();
110            point.x = "TASK_3      X: 5";
111            Console.WriteLine(point.x);
112
113            // Person
114            Person person = new Person();
115            person.age = "TASK_4      Age: 18";
116            person.Print();
117
118            // Table
119            Table table = new Table();
120            table.rows = "TASK_5      Rows: 3";
121            table.cols = "      Cols: 4";
122            table.Display();
123
124            // Manager
125            Manager manager = new Manager();
126            manager.age = "TASK_6      Age: 30";
127            manager.name = "      Name: Aleksei";
```

```
127
128            Console.WriteLine(manager.GetAge());
129            Console.WriteLine(manager.GetName());
130
131            // Point3D
132            Point3D point3D = new Point3D();
133            point3D.x = "TASK_7      X: 1";
134            point3D.y = "      X: 2";
135            point3D.z = "      X: 3";
136            point3D.Show();
137
138            // Shop
139            Shop shop = new Shop();
140            shop.SetName("TASK_8      Shop");
141            Console.WriteLine(shop.GetName());
142
143        }
144    }
```

- Результат отладки:

```
C:\Users\User\Desktop\Основы > + | v
TASK_1      Name: Ivan
TASK_2      Year: 2020
TASK_3      X: 5
TASK_4      Age: 18
TASK_5      Rows: 3
              Cols: 4
TASK_6      Age: 30
              Name: Aleksei
TASK_7      X: 1
              X: 2
              X: 3
TASK_8      Shop
```

2. Конструктор класса

- Код:

```
1   using System;
2
3   namespace ConsoleApp1
4   {
5       // 2.1. Student с стандартным конструктором
6       // Ссылок 3
7       public class StudentC2
8       {
9           public string name;
10
11           // Ссылок 1
12           public StudentC2()
13           {
14               name = "Неизвестно";
15           }
16
17       // 2.2. Child с открытым стандартным конструктором
18       // Ссылок 5
19       public class ChildC2
20       {
21           public int age;
22
23           // Ссылок 2
24           public ChildC2()
25           {
26               age = 0;
27           }
28
29       // 2.3 и 2.4. Car с разными конструкторами
30       // Ссылок 8
31       public class CarC2
32       {
33           public int year;
```

```

31     public string name;
32     public string color;
33
34     // Конструктор с годом выпуска
35     public CarC2(int y)
36     {
37         year = y;
38     }
39
40     // Конструктор с именем и цветом
41     public CarC2(string n, string c)
42     {
43         name = n;
44         color = c;
45     }
46
47
48     // 2.5. Product с защищенным полем и конструктором копирования
49     public class ProductC2
50     {
51         protected string name;
52
53         public ProductC2(string n)
54         {
55             name = n;
56         }
57
58         public ProductC2(ProductC2 other)
59         {
60             name = other.name;
61         }
62     }
63
64     // 2.6. Person с закрытым полем и стандартным конструктором
65     public class PersonC2
66     {
67         private int age;
68
69         public PersonC2()
70         {
71             age = 18;
72         }
73
74         public void Print()
75         {
76             Console.WriteLine($"Возраст: {age}");
77         }
78
79     }
80
81     // 2.7. Manager с закрытыми полями и конструктором копирования
82     public class ManagerC2
83     {
84         private int age;
85         private string name;
86
87         public ManagerC2(int a, string n)
88         {
89             age = a;
90             name = n;
91         }
92
93         public ManagerC2(ManagerC2 other)
94         {
95             age = other.age;
96             name = other.name;
97         }
98
99         public int GetAge() => age;
100        public string GetName() => name;
101
102    // Главная программа
103    internal class Program
104    {
105        static void Main(string[] args)
106        {
107            // 2.1. Student
108            StudentC2 student = new StudentC2();
109            Console.WriteLine($"Student name: {student.name}");
110
111            // 2.2. Child
112            ChildC2 child1 = new ChildC2();
113            ChildC2 child2 = new ChildC2();
114            Console.WriteLine($"Child1 age: {child1.age}, Child2 age: {child2.age}");
115
116            // 2.3 и 2.4. Car
117            CarC2 lada = new CarC2("LADA VESTA", "black");
118            CarC2 bmw = new CarC2("BMW X5", "white");
119            CarC2 carWithYear = new CarC2(2022);
120            Console.WriteLine($"Car lada: {lada.name}, {lada.color}");
121            Console.WriteLine($"Car bmw: {bmw.name}, {bmw.color}");
122
123            Console.WriteLine($"Car year: {carWithYear.year}");
124
125            // 2.5. Product
126            ProductC2 product1 = new ProductC2("Товар1");
127            ProductC2 product2 = new ProductC2(product1); // копирование
128            Console.WriteLine("Создан объект Product с копированием");
129
130            // 2.6. Person
131            PersonC2 person = new PersonC2();
132            person.Print(); // Выводится: Возраст: 18
133
134            // 2.7. Manager
135            ManagerC2 manager1 = new ManagerC2(20, "Дамир");
136            ManagerC2 manager2 = new ManagerC2(18, "Сара");
137            Console.WriteLine($"{manager1.GetName()} - {manager1.GetAge()}");
138            Console.WriteLine($"{manager2.GetName()} - {manager2.GetAge()}");
139        }
140    }

```

```

61     }
62
63
64     // 2.6. Person с закрытым полем и стандартным конструктором
65     public class PersonC2
66     {
67         private int age;
68
69         public PersonC2()
70         {
71             age = 18;
72         }
73
74         public void Print()
75         {
76             Console.WriteLine($"Возраст: {age}");
77         }
78
79     }
80
81     // 2.7. Manager с закрытыми полями и конструктором копирования
82     public class ManagerC2
83     {
84         private int age;
85         private string name;
86
87         public ManagerC2(int a, string n)
88         {
89             age = a;
90             name = n;
91         }
92
93         public ManagerC2(ManagerC2 other)
94         {
95             age = other.age;
96             name = other.name;
97         }
98
99         public int GetAge() => age;
100        public string GetName() => name;
101
102    // Главная программа
103    internal class Program
104    {
105        static void Main(string[] args)
106        {
107            // 2.1. Student
108            StudentC2 student = new StudentC2();
109            Console.WriteLine($"Student name: {student.name}");
110
111            // 2.2. Child
112            ChildC2 child1 = new ChildC2();
113            ChildC2 child2 = new ChildC2();
114            Console.WriteLine($"Child1 age: {child1.age}, Child2 age: {child2.age}");
115
116            // 2.3 и 2.4. Car
117            CarC2 lada = new CarC2("LADA VESTA", "black");
118            CarC2 bmw = new CarC2("BMW X5", "white");
119            CarC2 carWithYear = new CarC2(2022);
120            Console.WriteLine($"Car lada: {lada.name}, {lada.color}");
121            Console.WriteLine($"Car bmw: {bmw.name}, {bmw.color}");
122
123            Console.WriteLine($"Car year: {carWithYear.year}");
124
125            // 2.5. Product
126            ProductC2 product1 = new ProductC2("Товар1");
127            ProductC2 product2 = new ProductC2(product1); // копирование
128            Console.WriteLine("Создан объект Product с копированием");
129
130            // 2.6. Person
131            PersonC2 person = new PersonC2();
132            person.Print(); // Выводится: Возраст: 18
133
134            // 2.7. Manager
135            ManagerC2 manager1 = new ManagerC2(20, "Дамир");
136            ManagerC2 manager2 = new ManagerC2(18, "Сара");
137            Console.WriteLine($"{manager1.GetName()} - {manager1.GetAge()}");
138            Console.WriteLine($"{manager2.GetName()} - {manager2.GetAge()}");
139        }
140    }

```

```

92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140

```

```

122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140

```

- Результат отладки:

```
Консоль отладки Microsoft V... + ▾
Student name: Неизвестно
Child1 age: 0, Child2 age: 0
Car lada: LADA VESTA, black
Car bmw: BMW X5, white
Car year: 2022
Создан объект Product с копированием
Возраст: 18
Дамир - 20
Сара - 18
```

3. Свойства класса

- Код:

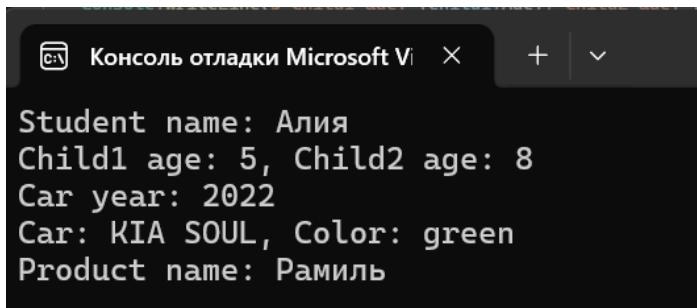
```
1  using System;
2
3  namespace ConsoleApp1
4  {
5      // 3.1. Класс Student со строковым автосвойством Name
6      public class Student
7      {
8          public string Name { get; set; }
9      }
10
11     // 3.2. Класс Child с открытым стандартным конструктором
12     public class Child
13     {
14         public int Age { get; set; }
15
16         public Child()
17         {
18         }
19
20
21     // 3.3. Класс Car с полем и свойством с проверкой
22     public class Car
23     {
24         private int year; // закрытое поле
25
26         public int Year // открытое свойство
27         {
28             get { return year; }
29
30             set
31             {
32                 if (value > 0) // разрешены только положительные числа
33                 {
34                     year = value;
35                 }
36             }
37
38             public Car() // стандартный конструктор
39             {
40             }
41
42
43     // 3.4. Класс Car2 с автосвойствами
44     public class Car2
45     {
46         public string Name { get; set; }
47         public string Color { get; set; }
48
49         public Car2()
50         {
51         }
52
53
54     // 3.5. Класс Product с защищенным полем и свойством только для чтения
55     public class Product
56     {
57         protected string name;
```

```

59     public string Name
60     {
61         get { return name; } // только чтение
62     }
63
64     Ссылка:1
65     public Product()
66     {
67         name = "Рамиль";
68     }
69
70     // Главная программа
71     Ссылка:0
72     internal class Program
73     {
74         Ссылка:0
75         static void Main(string[] args)
76         {
77             // 3.1. Student
78             Student student = new Student();
79             student.Name = "Алия";
80             Console.WriteLine($"Student name: {student.Name}");
81
82             // 3.2. Child (создание двух экземпляров с инициализацией)
83             Child child1 = new Child { Age = 5 };
84             Child child2 = new Child { Age = 8 };
85             Console.WriteLine($"Child1 age: {child1.Age}, Child2 age: {child2.Age}");
86
87             // 3.3. Car с проверкой значения Year
88             Car car = new Car();
89             car.Year = 2022;
90             Console.WriteLine($"Car year: {car.Year}");
91
92             // 3.4. Car2 с объектной инициализацией
93             Car2 kia = new Car2
94             {
95                 Name = "KIA SOUL",
96                 Color = "green"
97             };
98             Console.WriteLine($"Car: {kia.Name}, Color: {kia.Color}");
99
100            // 3.5. Product
101            Product product = new Product();
102            Console.WriteLine($"Product name: {product.Name}");
103        }
104    }

```

- Результат отладки:



Консоль отладки Microsoft Visual Studio

```

Student name: Алия
Child1 age: 5, Child2 age: 8
Car year: 2022
Car: KIA SOUL, Color: green
Product name: Рамиль

```

4. Наследование

- Код:

```
1  using System;
2
3  namespace ConsoleApp1
4  {
5      // 4.1. Person – базовый класс, Student – дочерний
6      public class Person
7      {
8          public string Name { get; set; }
9      }
10
11     public class Student : Person
12     {
13         public string University { get; set; }
14     }
15
16     // 4.2. Animal – родительский, Cat и Dog – дочерние
17     public class Animal
18     {
19         public virtual void Speak()
20         {
21             Console.WriteLine("Животное издает звук");
22         }
23     }
24
25     public class Cat : Animal
26     {
27         public override void Speak()
28         {
29             Console.WriteLine("Кошка говорит: Мяу");
30         }
31     }
32
33     public class Dog : Animal
34     {
35         public override void Speak()
36         {
37             Console.WriteLine("Собака говорит: Гав");
38         }
39     }
40
41     // 4.3. Entity – базовый, Product – производный
42     public class Entity
43     {
44         public int Id { get; set; }
45     }
46
47     public class Product : Entity
48     {
49         public string Name { get; set; }
50     }
51
52     // 4.4. Dishes – родительский, Cup – дочерний
53     public class Dishes
54     {
55         public string Material { get; set; }
56     }
57
58     public class Cup : Dishes
59     {
60         public double Volume { get; set; }
61     }
62
63     // 4.5. Entity → Staff → Manager (многоуровневое наследование)
64     public class EntityBase
65     {
66         public int Id { get; set; }
67     }
68
69     public class Staff : EntityBase
70     {
71         public string Position { get; set; }
72     }
73
74     public class Manager : Staff
75     {
76         public int Level { get; set; }
77     }
78
79     // 4.6. Animal – родительский, Predator – дочерний
80     public class Animal2
81     {
82         protected int age; // доступен в Animal2 и Predator
83     }
```

```
29
30     }
31 }
32
33 public class Dog : Animal
34 {
35     public override void Speak()
36     {
37         Console.WriteLine("Собака говорит: Гав");
38     }
39 }
40
41 // 4.3. Entity – базовый, Product – производный
42 public class Entity
43 {
44     public int Id { get; set; }
45 }
46
47 public class Product : Entity
48 {
49     public string Name { get; set; }
50 }
51
52 // 4.4. Dishes – родительский, Cup – дочерний
53 public class Dishes
54 {
55     public string Material { get; set; }
56 }
57
58 public class Cup : Dishes
59 {
60     public double Volume { get; set; }
61 }
62
63 // 4.5. Entity → Staff → Manager (многоуровневое наследование)
64 public class EntityBase
65 {
66     public int Id { get; set; }
67 }
68
69 public class Staff : EntityBase
70 {
71     public string Position { get; set; }
72 }
73
74 public class Manager : Staff
75 {
76     public int Level { get; set; }
77 }
78
79 // 4.6. Animal – родительский, Predator – дочерний
80 public class Animal2
81 {
82     protected int age; // доступен в Animal2 и Predator
83 }
```

```
58
59     public class Cup : Dishes
60     {
61         public double Volume { get; set; }
62     }
63
64     // 4.5. Entity → Staff → Manager (многоуровневое наследование)
65     public class EntityBase
66     {
67         public int Id { get; set; }
68     }
69
70     public class Staff : EntityBase
71     {
72         public string Position { get; set; }
73     }
74
75     public class Manager : Staff
76     {
77         public int Level { get; set; }
78     }
79
80     // 4.6. Animal – родительский, Predator – дочерний
81     public class Animal2
82     {
83         protected int age; // доступен в Animal2 и Predator
84     }
```

```
84
85
86
87
88
89     public Animal2(int a)
90     {
91         age = a;
92     }
93
94     Ссылка: 0
95     public int GetAge()
96     {
97         return age;
98     }
99
100
101
102
103
104
105
106
107 // 4.7. Transport – базовый, SpaceShuttle – дочерний
108
109     Ссылка: 1
110     public class Transport
111     {
112         private string name; // инкапсулированное поле
113
114
115
116
117
118         Ссылка: 2
119         public class SpaceShuttle : Transport
120         {
121             Ссылка: 1
122             public void Launch()
123             {
124                 Console.WriteLine($"Космический шаттл {Name} запущен!");
125             }
126
127
128         // Главная программа
129         Ссылка: 0
130         internal class Program
131         {
132             Ссылка: 0
133             static void Main(string[] args)
134             {
135                 // 4.1
136                 Student student = new Student
137                 {
138                     Name = "Анна",
139                     University = "РНУ"
140                 };
141                 Console.WriteLine($"{student.Name} учится в {student.University}");
142
143
144
145
146
147             // 4.3
148             Product product = new Product { Id = 1, Name = "Ноутбук" };
149             Console.WriteLine($"Product: {product.Id}, {product.Name}");
150
151
152             // 4.4
153             Cup cup = new Cup { Material = "Стекло", Volume = 0.3 };
154             Console.WriteLine($"Cup: {cup.Material}, {cup.Volume} л");
155
156
157             // 4.5
158             Manager manager = new Manager
159             {
160                 Id = 10,
161                 Position = "Руководитель",
162                 Level = 2
163             };
164             Console.WriteLine($"Manager ID: {manager.Id}, Position: {manager.Position}");
165
166
167             // 4.6
168             Predator predator = new Predator(5);
169             predator.ShowAge();
170
171
172             // 4.7
173             SpaceShuttle shuttle = new SpaceShuttle();
174             shuttle.Name = "Discovery";
175             shuttle.Launch();
176
177
178 }
```

- Результат отладки:

```

    Консоль отладки Microsoft Visual Studio
    Алия учится в КНУ
    Кошка говорит: Мяу
    Собака говорит: Гав
    Product: 1, Ноутбук
    Cup: Стекло, 0,3 л
    Manager ID: 10, Position: Руководитель, Level: 2
    Возраст хищника: 5
    Космический шаттл Discovery запущен!
  
```

5.Полиморфизм

- Код:

```

1  using System;
2
3  namespace ConsoleApp1
4  {
5      // 5.1. Strategy с виртуальным методом Display
6      public class Strategy
7      {
8          public virtual void Display()
9          {
10             Console.WriteLine("Strategy");
11         }
12     }
13
14     // 5.2. Weather с виртуальным методом Show
15     public class Weather
16     {
17         public virtual void Show()
18         {
19             Console.WriteLine("My Weather");
20         }
21     }
22
23     // 5.3. Strategy и ConservativeStrategy
24     public class ConservativeStrategy : Strategy
25     {
26         public override void Display()
27         {
28             Console.WriteLine("Conservative Strategy");
29         }
30     }
  
```

```

31
32     // 5.4. Animal и Cat
33     public class Animal
34     {
35         private string type;
36
37         public Animal()
38         {
39             type = "My Type";
40         }
41
42         public virtual void Print()
43         {
44             Console.WriteLine(type);
45         }
46     }
47
48     public class Cat : Animal
49     {
50         private int age;
51
52         public Cat()
53         {
54             age = 5;
55         }
56
57         public override void Print()
58         {
  
```

```

59         {
60             Console.WriteLine(age);
61         }
62     }
63 
// 5.5. Абстрактный класс Entity
Ссылка:1
64     public abstract class Entity
65     {
66         Ссылка:2
67         public abstract void Display();
68     }
69 
// 5.6. Product наследуется от Entity
Ссылка:2
70     public class Product : Entity
71     {
72         Ссылка:2
73         public override void Display()
74         {
75             Console.WriteLine("My Product");
76         }
77     }
78 
// 5.7. Интерфейс IPrintable
Ссылка:1
79     public interface IPrintable
80     {
81         Ссылка:2
82         void Display();
83     }
84 
Ссылка:2
85     public class ConsolePrinting : IPrintable
86     {
87         Ссылка:2
88         public void Display()
89         {
90             Console.WriteLine("My Console");
91         }
92     }
93 
// Главная программа
Ссылка:0
94     internal class Program
95     {
96         Ссылка:0
97         static void Main(string[] args)
98         {
99             // 5.1
100            Strategy strategy = new Strategy();
101            strategy.Display();
102 
103            // 5.2
104            Weather weather = new Weather();
105            weather.Show();
106 
107            // 5.3 (полиморфизм)
108            Strategy conservative = new ConservativeStrategy();
109            conservative.Display();
110 
111            // 5.4
112            Animal animal = new Animal();
113            Animal cat = new Cat();
114 
115            animal.Print();
116            cat.Print();
117 
// 5.6 (абстрактный класс)
118            Product product = new Product();
119 
120            product.Display();
121 
// 5.7 (интерфейс)
122            ConsolePrinting printer = new ConsolePrinting();
123            printer.Display();
124        }
125    }

```

```

86     public void Display()
87     {
88         Console.WriteLine("My Console");
89     }
90 }
91 
// Главная программа
Ссылка:0
92     internal class Program
93     {
94         Ссылка:0
95         static void Main(string[] args)
96         {
97             // 5.1
98             Strategy strategy = new Strategy();
99             strategy.Display();
100 
101            // 5.2
102            Weather weather = new Weather();
103            weather.Show();
104 
105            // 5.3 (полиморфизм)
106            Strategy conservative = new ConservativeStrategy();
107            conservative.Display();
108 
109            // 5.4
110            Animal animal = new Animal();
111            Animal cat = new Cat();
112 
113            animal.Print();
114            cat.Print();
115 
// 5.6 (абстрактный класс)
116            Product product = new Product();
117 
118            product.Display();
119 
// 5.7 (интерфейс)
120            ConsolePrinting printer = new ConsolePrinting();
121            printer.Display();
122        }
123    }

```

```

118            product.Display();
119 
// 5.7 (интерфейс)
120            ConsolePrinting printer = new ConsolePrinting();
121            printer.Display();
122        }
123    }

```

- Результат отладки:

```

Strategy
My Weather
Conservative Strategy
My Type
5
My Product
My Console

```