



**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
33     // 2.5 Table
34     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
46     // 2.6 Manager
47     class Manager
48     {
49         public string age;
50         public string name;
51
52         // Ссылка 1
53         public string GetAge()
54         {
55             return age;
56         }
57
58         // Ссылка 1
59         public string GetName()
60         {
61             return name;
62         }
63     }
64
65     // 2.7 Point3D
```

```

63     class Point3D
64     {
65         public string x;
66         public string y;
67         public string z;
68
69         Ссылка: 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         class Shop
79         {
80             public string name;
81
82             Ссылка: 1
83             public string GetName()
84             {
85                 return name;
86             }
87
88             Ссылка: 1
89             public void SetName(string newName)
90             {
91                 name = newName;
92             }
93         }
94     }
95
96     Ссылка: 0

```

```

93     class Program
94     {
95         Ссылка: 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             Console.WriteLine(manager.GetAge());
128             Console.WriteLine(manager.GetName());
129
130             // Point3D
131             Point3D point3D = new Point3D();
132             point3D.x = "TASK_7      X: 1";
133             point3D.y = "      X: 2";
134             point3D.z = "      X: 3";
135             point3D.Show();
136
137             // Shop
138             Shop shop = new Shop();
139             shop.SetName("TASK_8      Shop");
140             Console.WriteLine(shop.GetName());
141
142             Console.ReadLine();
143         }
144     }

```

- **Результат отладки:**

```
C:\Users\User\Desktop\Основ  X + 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      // Ссылка: 3
7      public class StudentC2
8      {
9          public string name;
10
11         // Ссылка: 1
12         public StudentC2()
13         {
14             name = "Неизвестно";
15         }
16
17         // 2.2. Child с открытым стандартным конструктором
18         // Ссылка: 5
19         public class ChildC2
20         {
21             public int age;
22
23             // Ссылка: 2
24             public ChildC2()
25             {
26                 age = 0;
27             }
28
29         // 2.3 и 2.4. Car с разными конструкторами
30         // Ссылка: 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     Ссылка: 1
36     public CarC2(int y)
37     {
38         year = y;
39     }
40
41     // Конструктор с именем и цветом
42     Ссылка: 2
43     public CarC2(string n, string c)
44     {
45         name = n;
46         color = c;
47     }
48
49     // 2.5. Product с защищенным полем и конструктором копирования
50     Ссылка: 7
51     public class ProductC2
52     {
53         protected string name;
54
55         Ссылка: 1
56         public ProductC2(string n)
57         {
58             name = n;
59         }
60
61         Ссылка: 1
62         public ProductC2(ProductC2 other)
63         {
64             name = other.name;
65         }
66     }

```

```

61     }
62 }
63
64 // 2.6. Person с закрытым полем и стандартным конструктором
65 Ссылка: 3
66 public class PersonC2
67 {
68     private int age;
69
70     Ссылка: 1
71     public PersonC2()
72     {
73         age = 18;
74     }
75
76     Ссылка: 1
77     public void Print()
78     {
79         Console.WriteLine($"Возраст: {age}");
80     }
81
82     // 2.7. Manager с закрытыми полями и конструктором копирования
83     Ссылка: 7
84     public class ManagerC2
85     {
86         private int age;
87         private string name;
88
89         Ссылка: 2
90         public ManagerC2(int a, string n)
91         {
92             age = a;
93             name = n;
94         }
95
96         public ManagerC2(ManagerC2 other)
97         {
98             age = other.age;
99             name = other.name;
100         }
101
102         Ссылка: 2
103         public int GetAge() => age;
104         Ссылка: 2
105         public string GetName() => name;
106     }
107
108 // Главная программа
109 Ссылка: 0
110 internal class Program
111 {
112     Ссылка: 0
113     static void Main(string[] args)
114     {
115         // 2.1. Student
116         StudentC2 student = new StudentC2();
117         Console.WriteLine($"Student name: {student.name}");
118
119         // 2.2. Child
120         ChildC2 child1 = new ChildC2();
121         ChildC2 child2 = new ChildC2();
122         Console.WriteLine($"Child1 age: {child1.age}, Child2 age: {child2.age}");
123
124         // 2.3 и 2.4. Car
125         CarC2 lada = new CarC2("LADA VESTA", "black");
126         CarC2 bmw = new CarC2("BMW X5", "white");
127         CarC2 carWithYear = new CarC2(2022);
128         Console.WriteLine($"Car lada: {lada.name}, {lada.color}");
129         Console.WriteLine($"Car bmw: {bmw.name}, {bmw.color}");
130
131         Console.WriteLine($"Car year: {carWithYear.year}");
132
133         // 2.5. Product
134         ProductC2 product1 = new ProductC2("Товар1");
135         ProductC2 product2 = new ProductC2(product1); // копирование
136         Console.WriteLine($"Создан объект Product с копированием");
137
138         // 2.6. Person
139         PersonC2 person = new PersonC2();
140         person.Print(); // Выведет: Возраст: 18
141
142         // 2.7. Manager
143         ManagerC2 manager1 = new ManagerC2(20, "Дэмп");
144         ManagerC2 manager2 = new ManagerC2(18, "Сапа");
145         Console.WriteLine($"manager1.GetName() - {manager1.GetAge()}");
146         Console.WriteLine($"manager2.GetName() - {manager2.GetAge()}");
147     }
148 }

```

```

92     public ManagerC2(ManagerC2 other)
93     {
94         age = other.age;
95         name = other.name;
96     }
97
98     Ссылка: 2
99     public int GetAge() => age;
100     Ссылка: 2
101     public string GetName() => name;
102 }
103
104 // Главная программа
105 Ссылка: 0
106 internal class Program
107 {
108     Ссылка: 0
109     static void Main(string[] args)
110     {
111         // 2.1. Student
112         StudentC2 student = new StudentC2();
113         Console.WriteLine($"Student name: {student.name}");
114
115         // 2.2. Child
116         ChildC2 child1 = new ChildC2();
117         ChildC2 child2 = new ChildC2();
118         Console.WriteLine($"Child1 age: {child1.age}, Child2 age: {child2.age}");
119
120         // 2.3 и 2.4. Car
121         CarC2 lada = new CarC2("LADA VESTA", "black");
122         CarC2 bmw = new CarC2("BMW X5", "white");
123         CarC2 carWithYear = new CarC2(2022);
124         Console.WriteLine($"Car lada: {lada.name}, {lada.color}");
125         Console.WriteLine($"Car bmw: {bmw.name}, {bmw.color}");
126
127         Console.WriteLine($"Car year: {carWithYear.year}");
128
129         // 2.5. Product
130         ProductC2 product1 = new ProductC2("Товар1");
131         ProductC2 product2 = new ProductC2(product1); // копирование
132         Console.WriteLine($"Создан объект Product с копированием");
133
134         // 2.6. Person
135         PersonC2 person = new PersonC2();
136         person.Print(); // Выведет: Возраст: 18
137
138         // 2.7. Manager
139         ManagerC2 manager1 = new ManagerC2(20, "Дэмп");
140         ManagerC2 manager2 = new ManagerC2(18, "Сапа");
141         Console.WriteLine($"manager1.GetName() - {manager1.GetAge()}");
142         Console.WriteLine($"manager2.GetName() - {manager2.GetAge()}");
143     }
144 }

```

```

122     Console.WriteLine($"Car year: {carWithYear.year}");
123
124     // 2.5. Product
125     ProductC2 product1 = new ProductC2("Товар1");
126     ProductC2 product2 = new ProductC2(product1); // копирование
127     Console.WriteLine($"Создан объект Product с копированием");
128
129     // 2.6. Person
130     PersonC2 person = new PersonC2();
131     person.Print(); // Выведет: Возраст: 18
132
133     // 2.7. Manager
134     ManagerC2 manager1 = new ManagerC2(20, "Дэмп");
135     ManagerC2 manager2 = new ManagerC2(18, "Сапа");
136     Console.WriteLine($"manager1.GetName() - {manager1.GetAge()}");
137     Console.WriteLine($"manager2.GetName() - {manager2.GetAge()}");
138 }
139
140 }
141
142 }
143
144 }
145
146 }
147
148 }

```

- **Результат отладки:**

```

Консоль отладки Microsoft Vi  X  +  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
Сапа - 18

```

3.Свойства класса

- **Код:**

```

1  using System;
2
3  namespace ConsoleApp1
4  {
5      // 3.1. Класс Student со строковым атрибутом Name
6      // Ссылка 2
7      public class Student
8      {
9          // Ссылка 2
10         public string Name { get; set; }
11     }
12
13     // 3.2. Класс Child с открытым стандартным конструктором
14     // Ссылка 5
15     public class Child
16     {
17         // Ссылка 4
18         public int Age { get; set; }
19
20         // Ссылка 2
21         public Child()
22         {
23         }
24     }
25
26     // 3.3. Класс Car с полем и свойством с проверкой
27     // Ссылка 3
28     public class Car
29     {
30         private int year; // закрытое поле
31
32         // Ссылка 2
33         public int Year // открытое свойство
34         {
35             get { return year; }
36         }
37     }
38 }

```

```

39         set
40         {
41             if (value > 0) // разрешены только положительные числа
42             {
43                 year = value;
44             }
45         }
46     }
47
48     // Ссылка 1
49     public Car() // стандартный конструктор
50     {
51     }
52
53     // 3.4. Класс Car2 с атрибутами
54     // Ссылка 3
55     public class Car2
56     {
57         // Ссылка 2
58         public string Name { get; set; }
59         // Ссылка 2
60         public string Color { get; set; }
61
62         // Ссылка 1
63         public Car2()
64         {
65         }
66     }
67
68     // 3.5. Класс Product с защищенным полем и свойством только для чтения
69     // Ссылка 3
70     public class Product
71     {
72         protected string name;
73     }
74 }

```

```

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             // 3.4. Car2 с объектной инициализацией
92             Car2 kia = new Car2
93             {
94                 Name = "KIA SOUL",
95                 Color = "green"
96             };
97             Console.WriteLine($"Car: {kia.Name}, Color: {kia.Color}");
98
99             // 3.5. Product
100             Product product = new Product();
101             Console.WriteLine($"Product name: {product.Name}");
102         }
103     }

```

- Результат отладки:

```

Консоль отладки Microsoft V
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     // Ссылка 1
7     public class Person
8     {
9         // Ссылка 2
10        public string Name { get; set; }
11    }
12
13    // Ссылка 2
14    public class Student : Person
15    {
16        // Ссылка 2
17        public string University { get; set; }
18    }
19
20    // 4.2. Animal - родительский, Cat и Dog - дочерние
21    // Ссылка 4
22    public class Animal
23    {
24        // Ссылка 4
25        public virtual void Speak()
26        {
27            Console.WriteLine("Животное издает звук");
28        }
29    }
30
31    // Ссылка 1
32    public class Cat : Animal
33    {
34        // Ссылка 3
35        public override void Speak()
36        {
37            Console.WriteLine("Кошка говорит: Мяу");
38        }
39    }
40
41    // Ссылка 3
42    public class Dog : Animal
43    {
44        // Ссылка 3
45        public override void Speak()
46        {
47            Console.WriteLine("Собака говорит: Гав");
48        }
49    }
50
51    // 4.3. Entity - базовый, Product - производный
52    // Ссылка 1
53    public class Entity
54    {
55        // Ссылка 2
56        public int Id { get; set; }
57    }
58
59    // Ссылка 2
60    public class Product : Entity
61    {
62        // Ссылка 2
63        public string Name { get; set; }
64    }
65
66    // 4.4. Dishes - родительский, Cup - дочерний
67    // Ссылка 1
68    public class Dishes
69    {
70        // Ссылка 2
71        public string Material { get; set; }
72    }
73
74    // Ссылка 2
75    public class Cup : Dishes
76    {
77        // Ссылка 2
78        public double Volume { get; set; }
79    }
80
81    // 4.5. Entity -> Staff -> Manager (многоуровневое наследование)
82    // Ссылка 1
83    public class EntityBase
84    {
85        // Ссылка 2
86        public int Id { get; set; }
87    }
88
89    // Ссылка 1
90    public class Staff : EntityBase
91    {
92        // Ссылка 2
93        public string Position { get; set; }
94    }
95
96    // Ссылка 2
97    public class Manager : Staff
98    {
99        // Ссылка 2
100       public int Level { get; set; }
101    }
102
103    // 4.6. Animal - родительский, Predator - дочерний
104    // Ссылка 3
105    public class Animal2
106    {
107        protected int age; // доступен в Animal2 и Predator
108    }
109
110    // Ссылка 3
111    public class Predator : Animal2
112    {
113        // Ссылка 3
114        public int HuntLevel { get; set; }
115    }
116
117    // Ссылка 3
118    public class Prey : Animal2
119    {
120        // Ссылка 3
121        public int RunSpeed { get; set; }
122    }
123
124    // Ссылка 3
125    public class Herbivore : Prey
126    {
127        // Ссылка 3
128        public string FoodType { get; set; }
129    }
130
131    // Ссылка 3
132    public class Carnivore : Prey
133    {
134        // Ссылка 3
135        public string FoodType { get; set; }
136    }
137
138    // Ссылка 3
139    public class Omnivore : Prey
140    {
141        // Ссылка 3
142        public string FoodType { get; set; }
143    }
144
145    // Ссылка 3
146    public class Invertebrate : Prey
147    {
148        // Ссылка 3
149        public string FoodType { get; set; }
150    }
151
152    // Ссылка 3
153    public class Vertebrate : Prey
154    {
155        // Ссылка 3
156        public string FoodType { get; set; }
157    }
158
159    // Ссылка 3
160    public class Mammal : Vertebrate
161    {
162        // Ссылка 3
163        public string FoodType { get; set; }
164    }
165
166    // Ссылка 3
167    public class Bird : Vertebrate
168    {
169        // Ссылка 3
170        public string FoodType { get; set; }
171    }
172
173    // Ссылка 3
174    public class Reptile : Vertebrate
175    {
176        // Ссылка 3
177        public string FoodType { get; set; }
178    }
179
180    // Ссылка 3
181    public class Amphibian : Vertebrate
182    {
183        // Ссылка 3
184        public string FoodType { get; set; }
185    }
186
187    // Ссылка 3
188    public class Fish : Vertebrate
189    {
190        // Ссылка 3
191        public string FoodType { get; set; }
192    }
193
194    // Ссылка 3
195    public class Invertebrate2 : Invertebrate
196    {
197        // Ссылка 3
198        public string FoodType { get; set; }
199    }
200
201    // Ссылка 3
202    public class Vertebrate2 : Vertebrate
203    {
204        // Ссылка 3
205        public string FoodType { get; set; }
206    }
207
208    // Ссылка 3
209    public class Mammal2 : Mammal
210    {
211        // Ссылка 3
212        public string FoodType { get; set; }
213    }
214
215    // Ссылка 3
216    public class Bird2 : Bird
217    {
218        // Ссылка 3
219        public string FoodType { get; set; }
220    }
221
222    // Ссылка 3
223    public class Reptile2 : Reptile
224    {
225        // Ссылка 3
226        public string FoodType { get; set; }
227    }
228
229    // Ссылка 3
230    public class Amphibian2 : Amphibian
231    {
232        // Ссылка 3
233        public string FoodType { get; set; }
234    }
235
236    // Ссылка 3
237    public class Fish2 : Fish
238    {
239        // Ссылка 3
240        public string FoodType { get; set; }
241    }
242
243    // Ссылка 3
244    public class Invertebrate3 : Invertebrate2
245    {
246        // Ссылка 3
247        public string FoodType { get; set; }
248    }
249
250    // Ссылка 3
251    public class Vertebrate3 : Vertebrate2
252    {
253        // Ссылка 3
254        public string FoodType { get; set; }
255    }
256
257    // Ссылка 3
258    public class Mammal3 : Mammal2
259    {
260        // Ссылка 3
261        public string FoodType { get; set; }
262    }
263
264    // Ссылка 3
265    public class Bird3 : Bird2
266    {
267        // Ссылка 3
268        public string FoodType { get; set; }
269    }
270
271    // Ссылка 3
272    public class Reptile3 : Reptile2
273    {
274        // Ссылка 3
275        public string FoodType { get; set; }
276    }
277
278    // Ссылка 3
279    public class Amphibian3 : Amphibian2
280    {
281        // Ссылка 3
282        public string FoodType { get; set; }
283    }
284
285    // Ссылка 3
286    public class Fish3 : Fish2
287    {
288        // Ссылка 3
289        public string FoodType { get; set; }
290    }
291
292    // Ссылка 3
293    public class Invertebrate4 : Invertebrate3
294    {
295        // Ссылка 3
296        public string FoodType { get; set; }
297    }
298
299    // Ссылка 3
300    public class Vertebrate4 : Vertebrate3
301    {
302        // Ссылка 3
303        public string FoodType { get; set; }
304    }
305
306    // Ссылка 3
307    public class Mammal4 : Mammal3
308    {
309        // Ссылка 3
310        public string FoodType { get; set; }
311    }
312
313    // Ссылка 3
314    public class Bird4 : Bird3
315    {
316        // Ссылка 3
317        public string FoodType { get; set; }
318    }
319
320    // Ссылка 3
321    public class Reptile4 : Reptile3
322    {
323        // Ссылка 3
324        public string FoodType { get; set; }
325    }
326
327    // Ссылка 3
328    public class Amphibian4 : Amphibian3
329    {
330        // Ссылка 3
331        public string FoodType { get; set; }
332    }
333
334    // Ссылка 3
335    public class Fish4 : Fish3
336    {
337        // Ссылка 3
338        public string FoodType { get; set; }
339    }
340
341    // Ссылка 3
342    public class Invertebrate5 : Invertebrate4
343    {
344        // Ссылка 3
345        public string FoodType { get; set; }
346    }
347
348    // Ссылка 3
349    public class Vertebrate5 : Vertebrate4
350    {
351        // Ссылка 3
352        public string FoodType { get; set; }
353    }
354
355    // Ссылка 3
356    public class Mammal5 : Mammal4
357    {
358        // Ссылка 3
359        public string FoodType { get; set; }
360    }
361
362    // Ссылка 3
363    public class Bird5 : Bird4
364    {
365        // Ссылка 3
366        public string FoodType { get; set; }
367    }
368
369    // Ссылка 3
370    public class Reptile5 : Reptile4
371    {
372        // Ссылка 3
373        public string FoodType { get; set; }
374    }
375
376    // Ссылка 3
377    public class Amphibian5 : Amphibian4
378    {
379        // Ссылка 3
380        public string FoodType { get; set; }
381    }
382
383    // Ссылка 3
384    public class Fish5 : Fish4
385    {
386        // Ссылка 3
387        public string FoodType { get; set; }
388    }
389
390    // Ссылка 3
391    public class Invertebrate6 : Invertebrate5
392    {
393        // Ссылка 3
394        public string FoodType { get; set; }
395    }
396
397    // Ссылка 3
398    public class Vertebrate6 : Vertebrate5
399    {
400        // Ссылка 3
401        public string FoodType { get; set; }
402    }
403
404    // Ссылка 3
405    public class Mammal6 : Mammal5
406    {
407        // Ссылка 3
408        public string FoodType { get; set; }
409    }
410
411    // Ссылка 3
412    public class Bird6 : Bird5
413    {
414        // Ссылка 3
415        public string FoodType { get; set; }
416    }
417
418    // Ссылка 3
419    public class Reptile6 : Reptile5
420    {
421        // Ссылка 3
422        public string FoodType { get; set; }
423    }
424
425    // Ссылка 3
426    public class Amphibian6 : Amphibian5
427    {
428        // Ссылка 3
429        public string FoodType { get; set; }
430    }
431
432    // Ссылка 3
433    public class Fish6 : Fish5
434    {
435        // Ссылка 3
436        public string FoodType { get; set; }
437    }
438
439    // Ссылка 3
440    public class Invertebrate7 : Invertebrate6
441    {
442        // Ссылка 3
443        public string FoodType { get; set; }
444    }
445
446    // Ссылка 3
447    public class Vertebrate7 : Vertebrate6
448    {
449        // Ссылка 3
450        public string FoodType { get; set; }
451    }
452
453    // Ссылка 3
454    public class Mammal7 : Mammal6
455    {
456        // Ссылка 3
457        public string FoodType { get; set; }
458    }
459
460    // Ссылка 3
461    public class Bird7 : Bird6
462    {
463        // Ссылка 3
464        public string FoodType { get; set; }
465    }
466
467    // Ссылка 3
468    public class Reptile7 : Reptile6
469    {
470        // Ссылка 3
471        public string FoodType { get; set; }
472    }
473
474    // Ссылка 3
475    public class Amphibian7 : Amphibian6
476    {
477        // Ссылка 3
478        public string FoodType { get; set; }
479    }
480
481    // Ссылка 3
482    public class Fish7 : Fish6
483    {
484        // Ссылка 3
485        public string FoodType { get; set; }
486    }
487
488    // Ссылка 3
489    public class Invertebrate8 : Invertebrate7
490    {
491        // Ссылка 3
492        public string FoodType { get; set; }
493    }
494
495    // Ссылка 3
496    public class Vertebrate8 : Vertebrate7
497    {
498        // Ссылка 3
499        public string FoodType { get; set; }
500    }
501
502    // Ссылка 3
503    public class Mammal8 : Mammal7
504    {
505        // Ссылка 3
506        public string FoodType { get; set; }
507    }
508
509    // Ссылка 3
510    public class Bird8 : Bird7
511    {
512        // Ссылка 3
513        public string FoodType { get; set; }
514    }
515
516    // Ссылка 3
517    public class Reptile8 : Reptile7
518    {
519        // Ссылка 3
520        public string FoodType { get; set; }
521    }
522
523    // Ссылка 3
524    public class Amphibian8 : Amphibian7
525    {
526        // Ссылка 3
527        public string FoodType { get; set; }
528    }
529
530    // Ссылка 3
531    public class Fish8 : Fish7
532    {
533        // Ссылка 3
534        public string FoodType { get; set; }
535    }
536
537    // Ссылка 3
538    public class Invertebrate9 : Invertebrate8
539    {
540        // Ссылка 3
541        public string FoodType { get; set; }
542    }
543
544    // Ссылка 3
545    public class Vertebrate9 : Vertebrate8
546    {
547        // Ссылка 3
548        public string FoodType { get; set; }
549    }
550
551    // Ссылка 3
552    public class Mammal9 : Mammal8
553    {
554        // Ссылка 3
555        public string FoodType { get; set; }
556    }
557
558    // Ссылка 3
559    public class Bird9 : Bird8
560    {
561        // Ссылка 3
562        public string FoodType { get; set; }
563    }
564
565    // Ссылка 3
566    public class Reptile9 : Reptile8
567    {
568        // Ссылка 3
569        public string FoodType { get; set; }
570    }
571
572    // Ссылка 3
573    public class Amphibian9 : Amphibian8
574    {
575        // Ссылка 3
576        public string FoodType { get; set; }
577    }
578
579    // Ссылка 3
580    public class Fish9 : Fish8
581    {
582        // Ссылка 3
583        public string FoodType { get; set; }
584    }
585
586    // Ссылка 3
587    public class Invertebrate10 : Invertebrate9
588    {
589        // Ссылка 3
590        public string FoodType { get; set; }
591    }
592
593    // Ссылка 3
594    public class Vertebrate10 : Vertebrate9
595    {
596        // Ссылка 3
597        public string FoodType { get; set; }
598    }
599
600    // Ссылка 3
601    public class Mammal10 : Mammal9
602    {
603        // Ссылка 3
604        public string FoodType { get; set; }
605    }
606
607    // Ссылка 3
608    public class Bird10 : Bird9
609    {
610        // Ссылка 3
611        public string FoodType { get; set; }
612    }
613
614    // Ссылка 3
615    public class Reptile10 : Reptile9
616    {
617        // Ссылка 3
618        public string FoodType { get; set; }
619    }
620
621    // Ссылка 3
622    public class Amphibian10 : Amphibian9
623    {
624        // Ссылка 3
625        public string FoodType { get; set; }
626    }
627
628    // Ссылка 3
629    public class Fish10 : Fish9
630    {
631        // Ссылка 3
632        public string FoodType { get; set; }
633    }
634
635    // Ссылка 3
636    public class Invertebrate11 : Invertebrate10
637    {
638        // Ссылка 3
639        public string FoodType { get; set; }
640    }
641
642    // Ссылка 3
643    public class Vertebrate11 : Vertebrate10
644    {
645        // Ссылка 3
646        public string FoodType { get; set; }
647    }
648
649    // Ссылка 3
650    public class Mammal11 : Mammal10
651    {
652        // Ссылка 3
653        public string FoodType { get; set; }
654    }
655
656    // Ссылка 3
657    public class Bird11 : Bird10
658    {
659        // Ссылка 3
660        public string FoodType { get; set; }
661    }
662
663    // Ссылка 3
664    public class Reptile11 : Reptile10
665    {
666        // Ссылка 3
667        public string FoodType { get; set; }
668    }
669
670    // Ссылка 3
671    public class Amphibian11 : Amphibian10
672    {
673        // Ссылка 3
674        public string FoodType { get; set; }
675    }
676
677    // Ссылка 3
678    public class Fish11 : Fish10
679    {
680        // Ссылка 3
681        public string FoodType { get; set; }
682    }
683
684    // Ссылка 3
685    public class Invertebrate12 : Invertebrate11
686    {
687        // Ссылка 3
688        public string FoodType { get; set; }
689    }
690
691    // Ссылка 3
692    public class Vertebrate12 : Vertebrate11
693    {
694        // Ссылка 3
695        public string FoodType { get; set; }
696    }
697
698    // Ссылка 3
699    public class Mammal12 : Mammal11
700    {
701        // Ссылка 3
702        public string FoodType { get; set; }
703    }
704
705    // Ссылка 3
706    public class Bird12 : Bird11
707    {
708        // Ссылка 3
709        public string FoodType { get; set; }
710    }
711
712    // Ссылка 3
713    public class Reptile12 : Reptile11
714    {
715        // Ссылка 3
716        public string FoodType { get; set; }
717    }
718
719    // Ссылка 3
720    public class Amphibian12 : Amphibian11
721    {
722        // Ссылка 3
723        public string FoodType { get; set; }
724    }
725
726    // Ссылка 3
727    public class Fish12 : Fish11
728    {
729        // Ссылка 3
730        public string FoodType { get; set; }
731    }
732
733    // Ссылка 3
734    public class Invertebrate13 : Invertebrate12
735    {
736        // Ссылка 3
737        public string FoodType { get; set; }
738    }
739
740    // Ссылка 3
741    public class Vertebrate13 : Vertebrate12
742    {
743        // Ссылка 3
744        public string FoodType { get; set; }
745    }
746
747    // Ссылка 3
748    public class Mammal13 : Mammal12
749    {
750        // Ссылка 3
751        public string FoodType { get; set; }
752    }
753
754    // Ссылка 3
755    public class Bird13 : Bird12
756    {
757        // Ссылка 3
758        public string FoodType { get; set; }
759    }
760
761    // Ссылка 3
762    public class Reptile13 : Reptile12
763    {
764        // Ссылка 3
765        public string FoodType { get; set; }
766    }
767
768    // Ссылка 3
769    public class Amphibian13 : Amphibian12
770    {
771        // Ссылка 3
772        public string FoodType { get; set; }
773    }
774
775    // Ссылка 3
776    public class Fish13 : Fish12
777    {
778        // Ссылка 3
779        public string FoodType { get; set; }
780    }
781
782    // Ссылка 3
783    public class Invertebrate14 : Invertebrate13
784    {
785        // Ссылка 3
786        public string FoodType { get; set; }
787    }
788
789    // Ссылка 3
790    public class Vertebrate14 : Vertebrate13
791    {
792        // Ссылка 3
793        public string FoodType { get; set; }
794    }
795
796    // Ссылка 3
797    public class Mammal14 : Mammal13
798    {
799        // Ссылка 3
800        public string FoodType { get; set; }
801    }
802
803    // Ссылка 3
804    public class Bird14 : Bird13
805    {
806        // Ссылка 3
807        public string FoodType { get; set; }
808    }
809
810    // Ссылка 3
811    public class Reptile14 : Reptile13
812    {
813        // Ссылка 3
814        public string FoodType { get; set; }
815    }
816
817    // Ссылка 3
818    public class Amphibian14 : Amphibian13
819    {
820        // Ссылка 3
821        public string FoodType { get; set; }
822    }
823
824    // Ссылка 3
825    public class Fish14 : Fish13
826    {
827        // Ссылка 3
828        public string FoodType { get; set; }
829    }
830
831    // Ссылка 3
832    public class Invertebrate15 : Invertebrate14
833    {
834        // Ссылка 3
835        public string FoodType { get; set; }
836    }
837
838    // Ссылка 3
839    public class Vertebrate15 : Vertebrate14
840    {
841        // Ссылка 3
842        public string FoodType { get; set; }
843    }
844
845    // Ссылка 3
846    public class Mammal15 : Mammal14
847    {
848        // Ссылка 3
849        public string FoodType { get; set; }
850    }
851
852    // Ссылка 3
853    public class Bird15 : Bird14
854    {
855        // Ссылка 3
856        public string FoodType { get; set; }
857    }
858
859    // Ссылка 3
860    public class Reptile15 : Reptile14
861    {
862        // Ссылка 3
863        public string FoodType { get; set; }
864    }
865
866    // Ссылка 3
867    public class Amphibian15 : Amphibian14
868    {
869        // Ссылка 3
870        public string FoodType { get; set; }
871    }
872
873    // Ссылка 3
874    public class Fish15 : Fish14
875    {
876        // Ссылка 3
877        public string FoodType { get; set; }
878    }
879
880    // Ссылка 3
881    public class Invertebrate16 : Invertebrate15
882    {
883        // Ссылка 3
884        public string FoodType { get; set; }
885    }
886
887    // Ссылка 3
888    public class Vertebrate16 : Vertebrate15
889    {
890        // Ссылка 3
891        public string FoodType { get; set; }
892    }
893
894    // Ссылка 3
895    public class Mammal16 : Mammal15
896    {
897        // Ссылка 3
898        public string FoodType { get; set; }
899    }
900
901    // Ссылка 3
902    public class Bird16 : Bird15
903    {
904        // Ссылка 3
905        public string FoodType { get; set; }
906    }
907
908    // Ссылка 3
909    public class Reptile16 : Reptile15
910    {
911        // Ссылка 3
912        public string FoodType { get; set; }
913    }
914
915    // Ссылка 3
916    public class Amphibian16 : Amphibian15
917    {
918        // Ссылка 3
919        public string FoodType { get; set; }
920    }
921
922    // Ссылка 3
923    public class Fish16 : Fish15
924    {
925        // Ссылка 3
926        public string FoodType { get; set; }
927    }
928
929    // Ссылка 3
930    public class Invertebrate17 : Invertebrate16
931    {
932        // Ссылка 3
933        public string FoodType { get; set; }
934    }
935
936    // Ссылка 3
937    public class Vertebrate17 : Vertebrate16
938    {
939        // Ссылка 3
940        public string FoodType { get; set; }
941    }
942
943    // Ссылка 3
944    public class Mammal17 : Mammal16
945    {
946        // Ссылка 3
947        public string FoodType { get; set; }
948    }
949
950    // Ссылка 3
951    public class Bird17 : Bird16
952    {
953        // Ссылка 3
954        public string FoodType { get; set; }
955    }
956
957    // Ссылка 3
958    public class Reptile17 : Reptile16
959    {
960        // Ссылка 3
961        public string FoodType { get; set; }
962    }
963
964    // Ссылка 3
965    public class Amphibian17 : Amphibian16
966    {
967        // Ссылка 3
968        public string FoodType { get; set; }
969    }
970
971    // Ссылка 3
972    public class Fish17 : Fish16
973    {
974        // Ссылка 3
975        public string FoodType { get; set; }
976    }
977
978    // Ссылка 3
979    public class Invertebrate18 : Invertebrate17
980    {
981        // Ссылка 3
982        public string FoodType { get; set; }
983    }
984
985    // Ссылка 3
986    public class Vertebrate18 : Vertebrate17
987    {
988        // Ссылка 3
989        public string FoodType { get; set; }
990    }
991
992    // Ссылка 3
993    public class Mammal18 : Mammal17
994    {
995        // Ссылка 3
996        public string FoodType { get; set; }
997    }
998
999    // Ссылка 3
1000   public class Bird18 : Bird17
1001   {
1002       // Ссылка 3
1003       public string FoodType { get; set; }
1004   }
1005
1006   // Ссылка 3
1007   public class Reptile18 : Reptile17
1008   {
1009       // Ссылка 3
1010       public string FoodType { get; set; }
1011   }
1012
1013   // Ссылка 3
1014   public class Amphibian18 : Amphibian17
1015   {
1016       // Ссылка 3
1017       public string FoodType { get; set; }
1018   }
1019
1020   // Ссылка 3
1021   public class Fish18 : Fish17
1022   {
1023       // Ссылка 3
1024       public string FoodType { get; set; }
1025   }
1026
1027   // Ссылка 3
1028   public class Invertebrate19 : Invertebrate18
1029   {
1030       // Ссылка 3
1031       public string FoodType { get; set; }
1032   }
1033
1034   // Ссылка 3
1035   public class Vertebrate19 : Vertebrate18
1036   {
1037       // Ссылка 3
1038       public string FoodType { get; set; }
1039   }
1040
1041   // Ссылка 3
1042   public class Mammal19 : Mammal18
1043   {
1044       // Ссылка 3
1045       public string FoodType { get; set; }
1046   }
1047
1048   // Ссылка 3
1049   public class Bird19 : Bird18
1050   {
1051       // Ссылка 3
1052       public string FoodType { get; set; }
1053   }
1054
1055   // Ссылка 3
1056   public class Reptile19 : Reptile18
1057   {
1058       // Ссылка 3
1059       public string FoodType { get; set; }
1060   }
1061
1062   // Ссылка 3
1063   public class Amphibian19 : Amphibian18
1064   {
1065       // Ссылка 3
1066       public string FoodType { get; set; }
1067   }
1068
1069   // Ссылка 3
1070   public class Fish19 : Fish18
1071   {
1072       // Ссылка 3
1073       public string FoodType { get; set; }
1074   }
1075
1076   // Ссылка 3
1077   public class Invertebrate20 : Invertebrate19
1078   {
1079       // Ссылка 3
1080       public string FoodType { get; set; }
1081   }
1082
1083   // Ссылка 3
1084   public class Vertebrate20 : Vertebrate19
1085   {
1086       // Ссылка 3
1087       public string FoodType { get; set; }
1088   }
1089
1090   // Ссылка 3
1091   public class Mammal20 : Mammal19
1092   {
1093       // Ссылка 3
1094       public string FoodType { get; set; }
1095   }
1096
1097   // Ссылка 3
1098   public class Bird20 : Bird19
1099   {
1100       // Ссылка 3
1101       public string FoodType { get; set; }
1102   }
1103
1104   // Ссылка 3
1105   public class Reptile20 : Reptile19
1106   {
1107       // Ссылка 3
1108       public string FoodType { get; set; }
1109   }
1110
1111   // Ссылка 3
1112   public class Amphibian20 : Amphibian19
1113   {
1114       // Ссылка 3
1115       public string FoodType { get; set; }
1116   }
1117
1118   // Ссылка 3
1119   public class Fish20 : Fish19
1120   {
1121       // Ссылка 3
1122       public string FoodType { get; set; }
1123   }
1124
1125   // Ссылка 3
1126   public class Invertebrate21 : Invertebrate20
1127   {
1128       // Ссылка 3
1129       public string FoodType { get; set; }
1130   }
1131
1132   // Ссылка 3
1133   public class Vertebrate21 : Vertebrate20
1134   {
1135       // Ссылка 3
1136       public string FoodType { get; set; }
1137   }
1138
1139   // Ссылка 3
1140   public class Mammal21 : Mammal20
1141   {
1142       // Ссылка 3
1143       public string FoodType { get; set; }
1144   }
1145
1146   // Ссылка 3
1147   public class Bird21 : Bird20
1148   {
1149       // Ссылка 3
1150       public string FoodType { get; set; }
1151   }
1152
1153   // Ссылка 3
1154   public class Reptile21 : Reptile20
1155   {
1156       // Ссылка 3
1157       public string FoodType { get; set; }
1158   }
1159
1160   // Ссылка 3
1161   public class Amphibian21 : Amphibian20
1162   {
1163       // Ссылка 3
1164       public string FoodType { get; set; }
1165   }
1166
1167   // Ссылка 3
1168   public class Fish21 : Fish20
1169   {
1170       // Ссылка 3
1171       public string FoodType { get; set; }
1172   }
1173
1174   // Ссылка 3
1175   public class Invertebrate22 : Invertebrate21
1176   {
1177       // Ссылка 3
1178       public string FoodType { get; set; }
1179   }
1180
1181   // Ссылка 3
1182   public class Vertebrate22 : Vertebrate21
1183   {
1184       // Ссылка 3
1185       public string FoodType { get; set; }
1186   }
1187
1188   // Ссылка 3
1189   public class Mammal22 : Mammal21
1190   {
1191       // Ссылка 3
1192       public string FoodType { get; set; }
1193   }
1194
1195   // Ссылка 3
1196   public class Bird22 : Bird21
1197   {
1198       // Ссылка 3
1199       public string FoodType { get; set; }
1200   }
1201
1202   // Ссылка 3
1203   public class Reptile22 : Reptile21
1204   {
1205       // Ссылка 3
1206       public string FoodType { get; set; }
1207   }
1208
1209   // Ссылка 3
1210   public class Amphibian22 : Amphibian21
1211   {
1212       // Ссылка 3
1213       public string FoodType { get; set; }
1214   }
1215
1216   // Ссылка 3
1217   public class Fish22 : Fish21
1218   {
1219       // Ссылка 3
1220       public string FoodType { get; set; }
1221   }
1222
1223   // Ссылка 3
1224   public class Invertebrate23 : Invertebrate22
1225   {
1226       // Ссылка 3
1227       public string FoodType { get; set; }
1228   }
1229
1230   // Ссылка 3
1231   public class Vertebrate23 : Vertebrate22
1232   {
1233       // Ссылка 3
1234       public string FoodType { get; set; }
1235   }
1236
1237   // Ссылка 3
1238   public class Mammal23 : Mammal22
1239   {
1240       // Ссылка 3
1241       public string FoodType { get; set; }
1242   }
1243
1244   // Ссылка 3
124
```

```

84     public Animal2(int a)
85     {
86         age = a;
87     }
88
89     Ссылка 0
90     public int GetAge()
91     {
92         return age;
93     }
94
95     Ссылка 3
96     public class Predator : Animal2
97     {
98         Ссылка 1
99         public Predator(int a) : base(a)
100         {
101         }
102
103         Ссылка 1
104         public void ShowAge()
105         {
106             Console.WriteLine($"Возраст хищника: {age}");
107         }
108     }
109
110     // 4.7. Transport - базовый, SpaceShuttle - дочерний
111     Ссылка 1
112     public class Transport
113     {
114         private string name; // инкапсулированное поле

```

```

112     public string Name
113     {
114         get { return name; }
115         set { name = value; }
116     }
117
118     Ссылка 2
119     public class SpaceShuttle : Transport
120     {
121         Ссылка 1
122         public void Launch()
123         {
124             Console.WriteLine($"Космический аппарат {Name} запущен!");
125         }
126     }
127
128     // Главная программа
129     Ссылка 0
130     internal class Program
131     {
132         Ссылка 0
133         static void Main(string[] args)
134         {
135             // 4.1
136             Student student = new Student
137             {
138                 Name = "Алиа",
139                 University = "КНУ"
140             };
141             Console.WriteLine($"{student.Name} учится в {student.University}");
142
143             // 4.2 (полиморфизм)
144             Animal cat = new Cat();

```

```

142             Animal dog = new Dog();
143             cat.Speak();
144             dog.Speak();
145
146             // 4.3
147             Product product = new Product { Id = 1, Name = "Ноутбук" };
148             Console.WriteLine($"Product: {product.Id}, {product.Name}");
149
150             // 4.4
151             Cup cup = new Cup { Material = "Стекло", Volume = 0.3 };
152             Console.WriteLine($"Cup: {cup.Material}, {cup.Volume} л");
153
154             // 4.5
155             Manager manager = new Manager
156             {
157                 Id = 10,
158                 Position = "Руководитель",
159                 Level = 2
160             };
161             Console.WriteLine($"Manager ID: {manager.Id}, Position: {manager.Position}");
162
163             // 4.6
164             Predator predator = new Predator(5);
165             predator.ShowAge();
166
167             // 4.7
168             SpaceShuttle shuttle = new SpaceShuttle();
169             shuttle.Name = "Discovery";
170             shuttle.Launch();
171         }
172     }
173 }

```

- **Результат отладки:**

```
Консоль отладки Microsoft V
Алия учится в КНУ
Кошка говорит: Мяу
Собака говорит: Гав
Product: 1, Ноутбук
Сир: 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         // 5.2. Weather с виртуальным методом Show
9         public virtual void Display()
10        {
11            Console.WriteLine("Strategy");
12        }
13    }
14
15    // 5.3. Strategy и ConservativeStrategy
16    public class Weather
17    {
18        public virtual void Show()
19        {
20            Console.WriteLine("My Weather");
21        }
22    }
23
24    // 5.3. Strategy и ConservativeStrategy
25    public class ConservativeStrategy : Strategy
26    {
27        public override void Display()
28        {
29            Console.WriteLine("Conservative Strategy");
30        }
31    }
32 }
```

```
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     }
```

```

59         Console.WriteLine(age);
60     }
61 }
62
63 // 5.5. Абстрактный класс Entity
64 Ссылка: 1 public abstract class Entity
65 {
66     Ссылка: 2 public abstract void Display();
67 }
68
69 // 5.6. Product наследуется от Entity
70 Ссылка: 2 public class Product : Entity
71 {
72     Ссылка: 2 public override void Display()
73     {
74         Console.WriteLine("My Product");
75     }
76 }
77
78 // 5.7. Интерфейс IPrintable
79 Ссылка: 1 public interface IPrintable
80 {
81     Ссылка: 2 void Display();
82 }
83
84 Ссылка: 2 public class ConsolePrinting : IPrintable
85 {
86     Ссылка: 3

```

```

86     public void Display()
87     {
88         Console.WriteLine("My Console");
89     }
90 }
91
92 // Главная программа
93 Ссылка: 0 internal class Program
94 {
95     Ссылка: 0 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
116         // 5.6 (абстрактный класс)
117         Product product = new Product();

```

```

118         product.Display();
119
120         // 5.7 (интерфейс)
121         ConsolePrinting printer = new ConsolePrinting();
122         printer.Display();
123     }
124 }
125

```

- **Результат отладки:**

```

Консоль отладки Microsoft Vi
Strategy
My Weather
Conservative Strategy
My Type
5
My Product
My Console

```