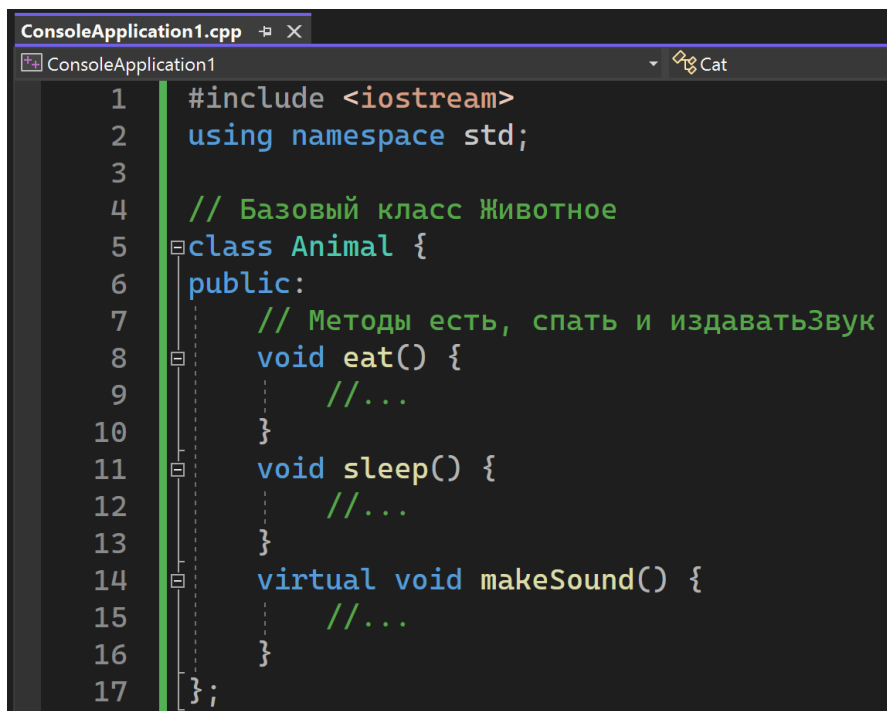


Цель домашнего задания: научиться создавать классы, наследовать классы, перегружать операторы, создавать шаблонные классы, работать с контейнерами стандартной библиотеки

Задание №2: Создайте базовый класс Животное с методами есть(), спать() и издаватьЗвук(). Затем создайте производные классы Кошка, Собака и Птица, которые наследуют от класса Животное и переопределяют метод издаватьЗвук() для каждого животного соответственно.

1. Создайте базовый класс Animal с методами eat, sleep и makeSound.



The screenshot shows a code editor window titled 'ConsoleApplication1.cpp'. The code defines a base class 'Animal' with three methods: 'eat()', 'sleep()', and 'makeSound()'. Each method has a placeholder comment '//...'. The code is as follows:

```
1  #include <iostream>
2  using namespace std;
3
4  // Базовый класс Животное
5  class Animal {
6  public:
7      // Методы есть, спать и издаватьЗвук
8      void eat() {
9          //...
10     }
11     void sleep() {
12         //...
13     }
14     virtual void makeSound() {
15         //...
16     }
17 };
```

2. Добавьте реализацию методов eat и sleep, которые выводят соответствующие сообщения на экран.

```
ConsoleApplication1.cpp  X
ConsoleApplication1  Cat

1  #include <iostream>
2  using namespace std;
3
4  // Базовый класс Животное
5  class Animal {
6  public:
7      // Методы есть, спать и издаватьЗвук
8      void eat() {
9          cout << "Animal is eating" << endl;
10     }
11     void sleep() {
12         cout << "Animal is sleeping" << endl;
13     }
14     virtual void makeSound() {
15         //...
16     }
17 };
```

3. Добавьте виртуальную реализацию метода makeSound, которая также выводит сообщение на экран.

```
ConsoleApplication1.cpp  X
ConsoleApplication1  Cat

1  #include <iostream>
2  using namespace std;
3
4  // Базовый класс Животное
5  class Animal {
6  public:
7      // Методы есть, спать и издаватьЗвук
8      void eat() {
9          cout << "Animal is eating" << endl;
10     }
11     void sleep() {
12         cout << "Animal is sleeping" << endl;
13     }
14     virtual void makeSound() {
15         cout << "Animal is making a sound" << endl;
16     }
17 };
```

4. Создайте производные классы Cat, Dog и Bird, которые наследуют от класса Animal.

```
19 // Производный класс Кошка
20 class Cat : public Animal {
21     |
22     };
23
24 // Производный класс Собака
25 class Dog : public Animal {
26     |
27     };
28
29 // Производный класс Птица
30 class Bird : public Animal {
31     |
32     };
```

5. Добавьте в каждый из производных классов переопределение метода `makeSound`, который выводит соответствующий звук животного на экран.

```

19 // Производный класс Кошка
20 class Cat : public Animal {
21 public:
22     // Переопределение метода издаватьЗвук для кошки
23     void makeSound() override {
24         cout << "Cat is meowing" << endl;
25     }
26 };
27
28 // Производный класс Собака
29 class Dog : public Animal {
30 public:
31     // Переопределение метода издаватьЗвук для собаки
32     void makeSound() override {
33         cout << "Dog is barking" << endl;
34     }
35 };
36
37 // Производный класс Птица
38 class Bird : public Animal {
39 public:
40     // Переопределение метода издаватьЗвук для птицы
41     void makeSound() override {
42         cout << "Bird is chirping" << endl;
43     }
44 };

```

6. В функции main создайте по одному объекту каждого класса.

```

46 int main() {
47     // Создание объектов каждого класса
48     Cat cat;
49     Dog dog;
50     Bird bird;

```

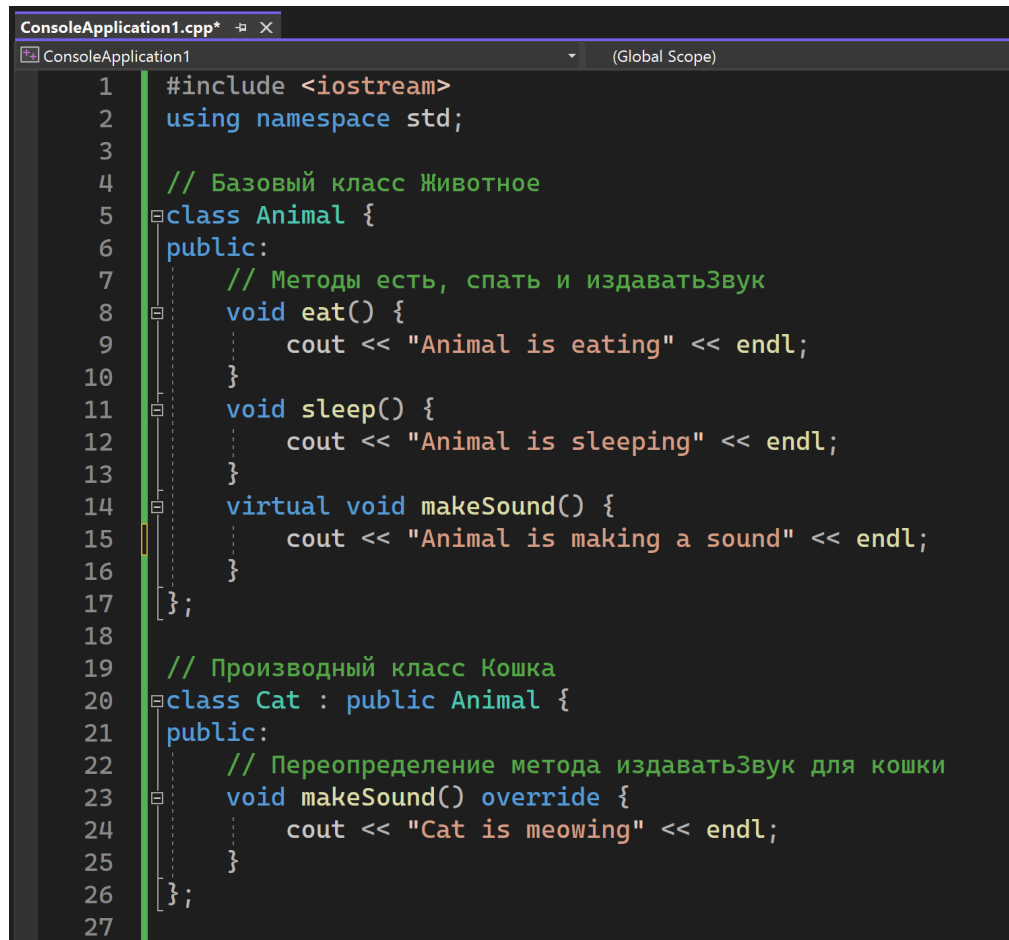
7. Вызовите метод makeSound для каждого из созданных объектов.

```

52 // Вызов метода издаватьЗвук для каждого объекта
53 cat.makeSound();
54 dog.makeSound();
55 bird.makeSound();
56
57 return 0;
58 }

```

8. У Вас должна получиться следующая программа:



```

ConsoleApplication1.cpp*
ConsoleApplication1 (Global Scope)
1  #include <iostream>
2  using namespace std;
3
4  // Базовый класс Животное
5  class Animal {
6  public:
7      // Методы есть, спать и издаватьЗвук
8      void eat() {
9          cout << "Animal is eating" << endl;
10     }
11     void sleep() {
12         cout << "Animal is sleeping" << endl;
13     }
14     virtual void makeSound() {
15         cout << "Animal is making a sound" << endl;
16     }
17 };
18
19 // Производный класс Кошка
20 class Cat : public Animal {
21 public:
22     // Переопределение метода издаватьЗвук для кошки
23     void makeSound() override {
24         cout << "Cat is meowing" << endl;
25     }
26 };
27

```

```

28 // Производный класс Собака
29 class Dog : public Animal {
30 public:
31     // Переопределение метода издаватьЗвук для собаки
32     void makeSound() override {
33         cout << "Dog is barking" << endl;
34     }
35 };
36
37 // Производный класс Птица
38 class Bird : public Animal {
39 public:
40     // Переопределение метода издаватьЗвук для птицы
41     void makeSound() override {
42         cout << "Bird is chirping" << endl;
43     }
44 };
45
46 int main() {
47     // Создание объектов каждого класса
48     Cat cat;
49     Dog dog;
50     Bird bird;
51
52     // Вызов метода издаватьЗвук для каждого объекта
53     cat.makeSound();
54     dog.makeSound();
55     bird.makeSound();
56
57     return 0;
58 }

```

Запустите программу и проверьте результат. Сделайте скриншот консоли и отправьте преподавателю.