|  |
| --- |
| class Person  {  public:      string name;      int age;      Person(string n, int a)      {          name = n; age = a;      }      void move()      {          cout << name << " is moving" << endl;      }  };  int main()  {      Person person = Person("Tom", 22);      cout << "Name: " << person.name << "\tAge: " << person.age << endl;      person.name = "Bob";      person.move();        return 0;  } |

Теперь в классе Person определен конструктор:

|  |
| --- |
| Person(string n, int a)  {      name = n; age = a;  } |

По сути конструктор представляет функцию, которая может принимать параметры и которая должна называться по имени класса. В данном случае конструктор принимает два параметра и передает их значения полям name и age.

Если в классе определены конструкторы, то при создании объекта этого класса необходимо вызвать один из его конструкторов.

|  |
| --- |
| Person person = Person("Tom", 22); |

После этого вызова у объекта person для поля name будет определено значение "Tom", а для поля age - значение 22. Вполедствии мы также сможем обращаться к этим полям и переустанавливать их значения.