#### 1. Использование умного указателя

##### **Что нужно сделать**

1. Реализуйте класс Dog, который использует стандартный умный указатель shared\_ptr для класса Toy.
2. Реализуйте у класса Dog метод getToy, с помощью которого собака подбирает игрушку.
   1. Если у собаки уже есть эта игрушка, необходимо вывести в консоль фразу: I already have this toy.
   2. Если в данный момент игрушка находится у другой собаки, нужно вывести в консоль: Another dog is playing with this toy.
   3. Если игрушка свободна, собака подбирает её.
3. Реализуйте у класса Dog метод dropToy, с помощью которого собака бросает игрушку на пол, после чего другая может подобрать её.
   1. Если у собаки в этот момент нет игрушки, в консоль необходимо вывести: Nothing to drop.

##### **Советы и рекомендации**

* Чтобы узнать, у скольких собак в данный момент есть игрушка, можно использовать метод use\_count() у класса shared\_ptr. При этом необходимо учитывать, что создание shared\_ptr от класса Toy тоже увеличивает показатели счётчика.
* Чтобы увидеть удаление класса Toy, используйте следующее определение:

class Toy {  
public:  
   Toy(const std::string& name) {  
       name\_ = name;  
   }  
   std::string getNmae() {  
       return name\_;  
   }  
   ~Toy() {  
       std::cout << "Toy " << name\_ << " was dropped " << std::endl;  
   }  
private:  
   std::string name\_;  
};