

# **Практическое занятие 3.20**

**Виртуальный деструктор**  
**Агрегация и композиция классов**

# **Виртуальный деструктор**

- 1. Зачем нужен?**
- 2. Зачем нужны «чисто виртуальные» методы?**
- 3. Может ли деструктор быть «чисто виртуальным»?**
- 4. Может ли быть виртуальным конструктор?**
- 5. Может ли деструктор вызывать другие виртуальные методы?**

# Агрегация

Ассоциация - это двустороннее семантическое отношение классов.

Агрегация - это форма ассоциации, показывающая связь между целым и его частью.

```
class Unit {
    std::string name;
    Part part;
public:
    Unit(Part& p) { // конструктор целого
        part = p; // Опасно?
    }
};

class Part {
    std::string name;
};
```

***При агрегации «целое» не управляет временем жизни своих «частей»***

# Композиция

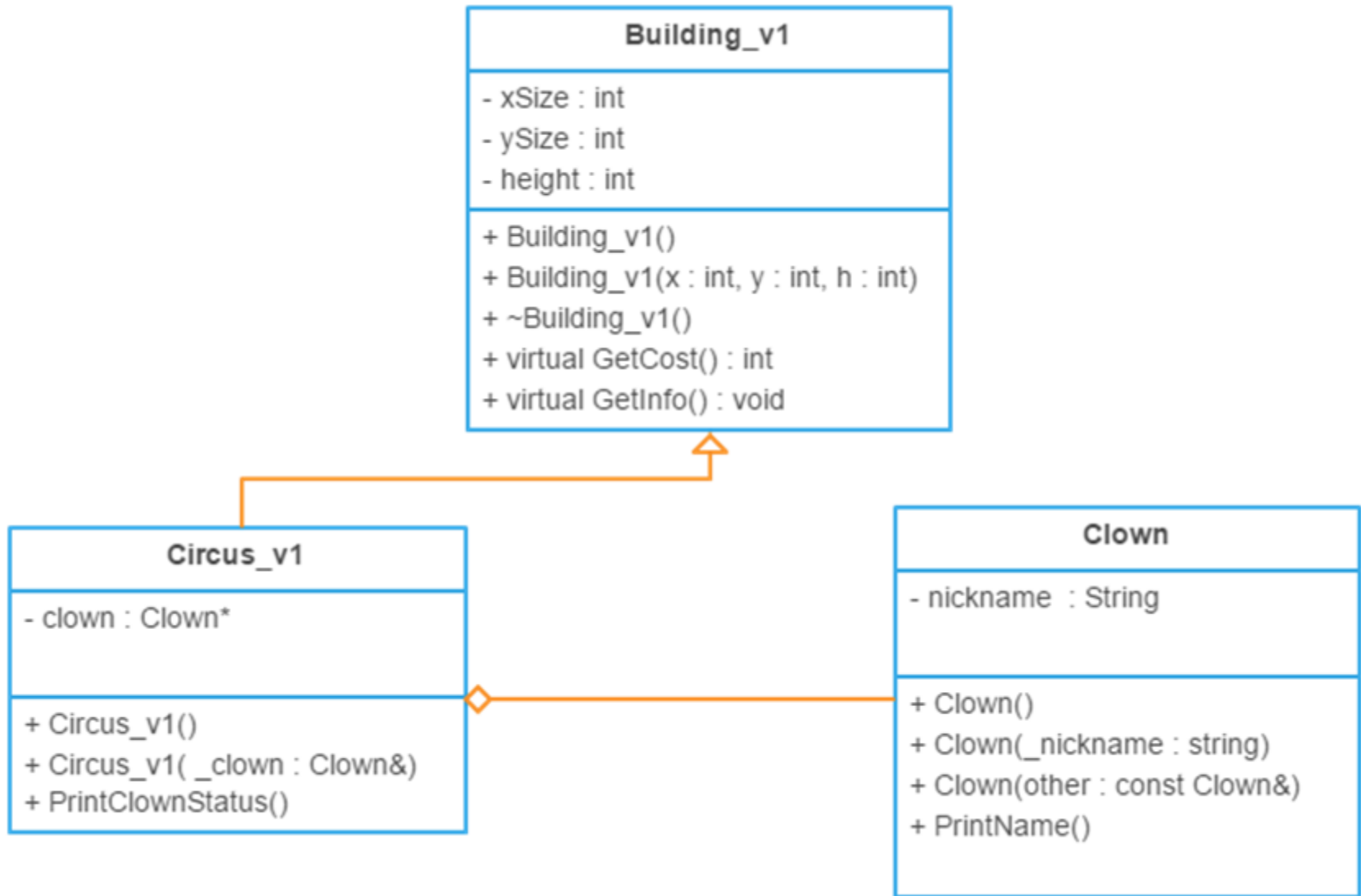
**Композиция (композиционная агрегация)** – это самая сильная форма ассоциации, где часть неотъемлема от единого целого.

Композиционная агрегация означает временную зависимость – создание целого, создание частей; удаление целого, удаление частей.

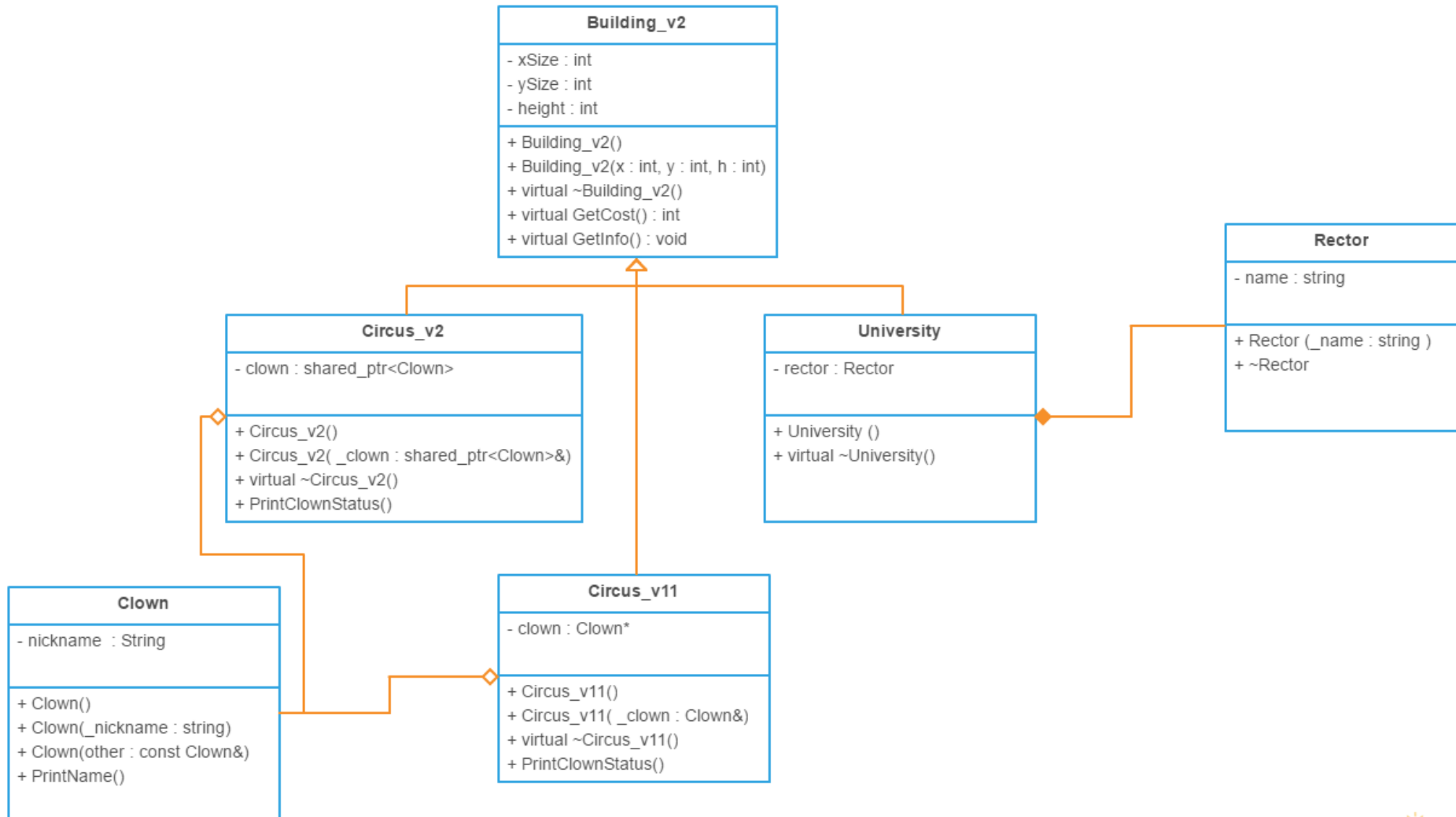
```
class Unit {  
    std::string name;  
    Part part;  
public:  
    Unit() : part() { // конструктор целого  
    }  
};  
class Part {  
    std::string name;  
};
```

***При композиции «целое» «управляет» временем жизни своих «частей»***

# Диаграмма классов примера v1



# Диаграмма классов примера v2



# Задача

Объявить класс **Point** – «точка на плоскости» и класс **Circle** – «круг с центром в точке», включающий экземпляр (объект) класса **Point** как поле для представления центра круга. Классы **Point** и **Circle** должны находиться в отношении **композиции**.

Члены класса **Point**: поля **X**, **Y** для координат точки; явно определенный конструктор с двумя параметрами – координатами точки; метод **distance()** для вычисления расстояния между двумя точками.

Члены класса **Circle**: поле **rad** – радиус окружности; поле **Point center** - центр круга; конструктор общего вида с тремя параметрами; метод для формирования строкового представления сведений об окружности.

В основной программе(в методе **main()** ) вводить данные для построения объектов класса **Circle** и, помещая их в **std::vector**, упорядочивать элементы массива по возрастанию произведения радиуса круга на удаление его центра от начала координат. Для сортировки элементов массива использовать метод **sort()**, второй параметр которого – **лямбда-выражение** для сравнения объектов типа **Circle**.