C++. Уровень 3

**Урок 4**

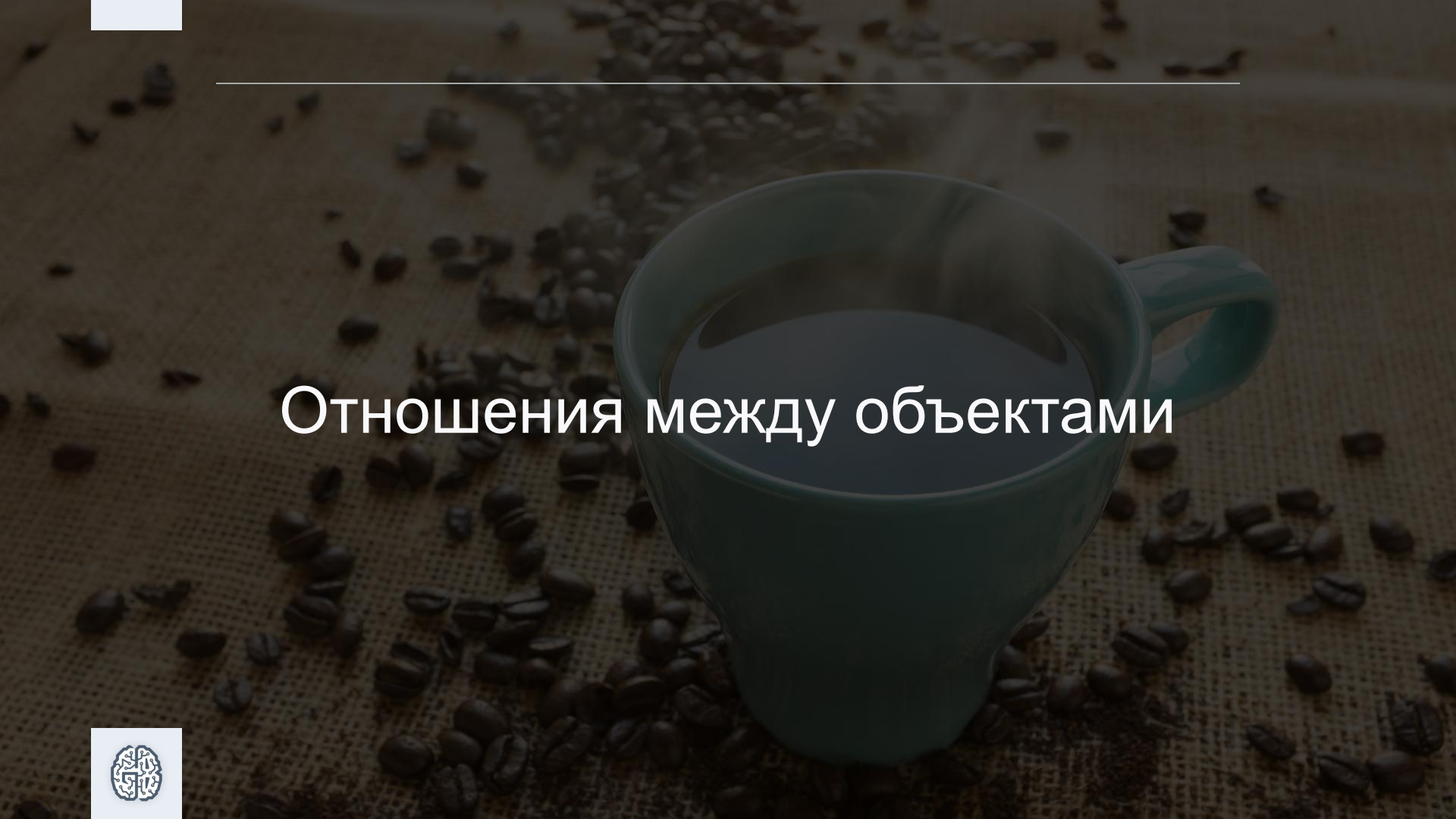
# Отношения между объектами

Композиция. Агрегация. Ассоциация. Зависимость. Контейнерные классы. Операторы приведения типа. Примеры программ на C++.

# План урока

- Композиция.
- Агрегация.
- Ассоциация.
- Зависимость.
- Контейнерные классы.
- Динамическое приведение типов.

# Отношения между объектами



# Разновидности отношений между объектами

# Композиция

Отношения между объектом и частью:

- часть (член) является составляющей объекта (класса);
- часть (член) может принадлежать только одному объекту (классу) в каждом конкретном случае;
- часть (член) существует под управлением объекта (класса);
- часть (член) не знает о существовании объекта (класса).

Стул имеет спинку и ножки

В комнате имеется пол, потолок и стены

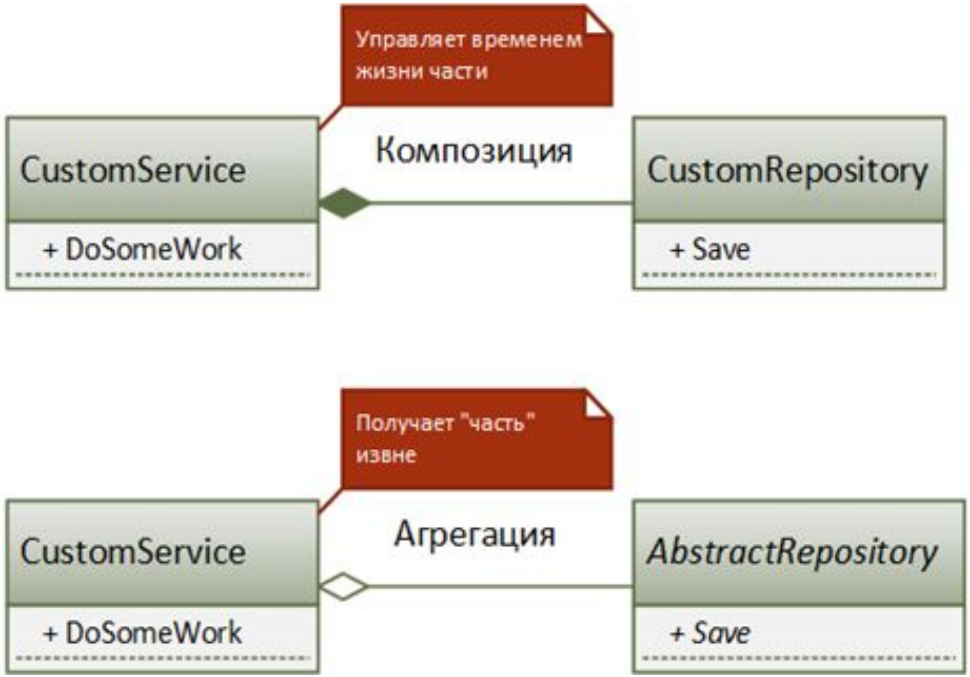
# Агрегация

Отношения между объектом и его частью:

- часть (член) является составляющей целого (класса);
- часть (член) может принадлежать более чем одному целому (классу) в каждом конкретном случае;
- часть (член) существует не под управлением целого (класса);
- часть (член) не знает о существовании целого (класса).

Некоторые объекты слабо взаимосвязаны
друг с другом, как компьютер и его периферийное оборудование

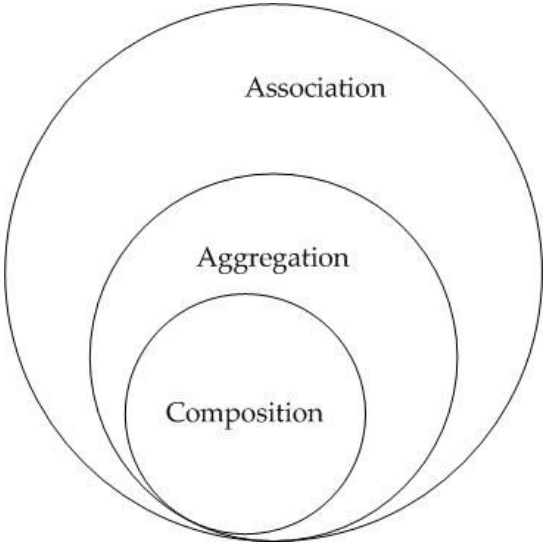
# Композиция vs Агрегация




# Ассоциация

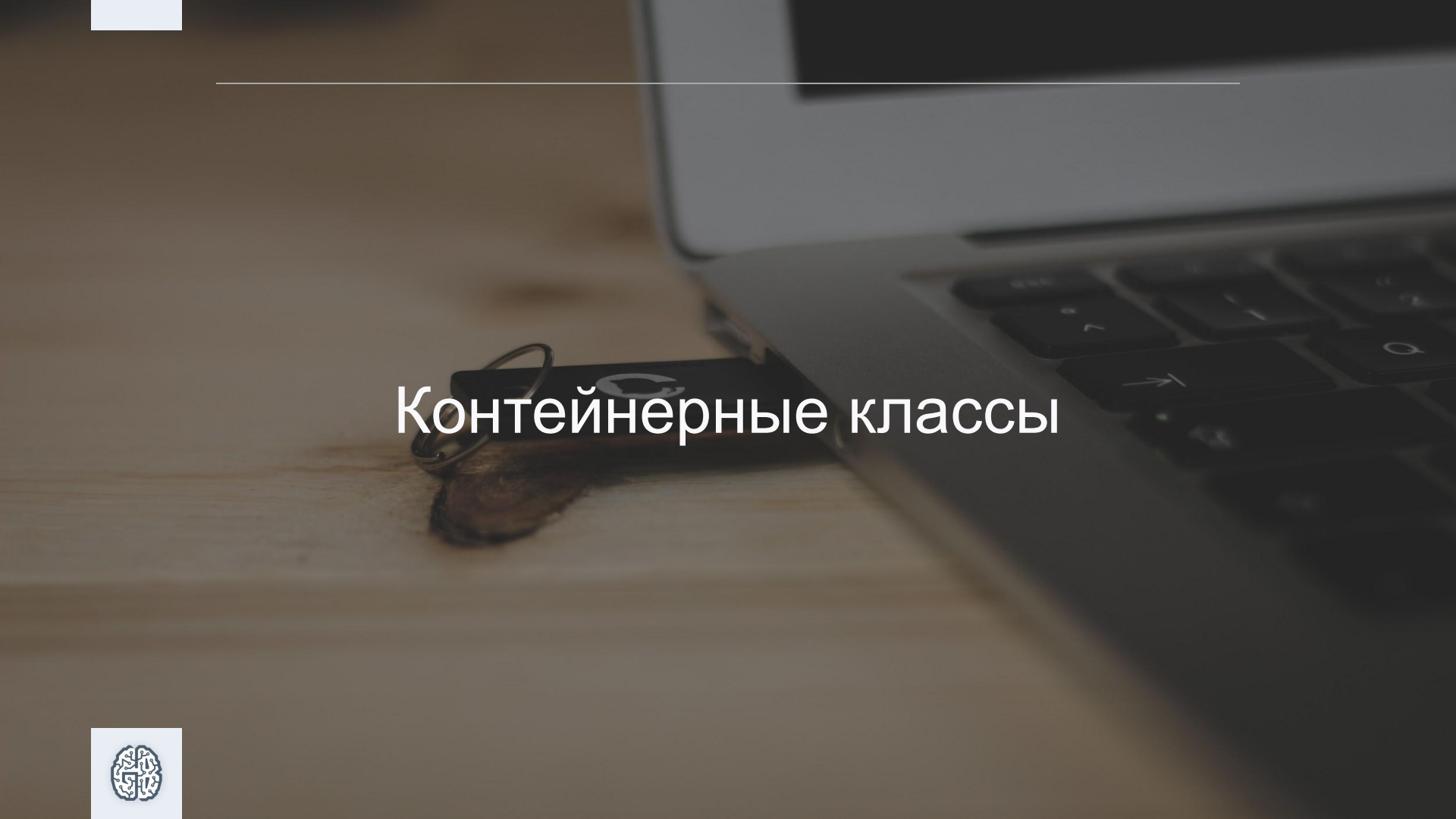
Отношения между объектом и его частью:

- первый объект (член) не связан со вторым объектом (классом);
- первый объект (член) может принадлежать одновременно сразу нескольким объектам (классам);
- первый объект (член) существует не под управлением второго объекта (класса);
- первый объект (член) может знать или не знать о существовании второго объекта (класса).




# Зависимость

Зависимость обозначает такое отношение между классами, когда изменение спецификации класса-поставщика может повлиять на работу зависимого класса, но не наоборот.



# Контейнерные классы

# Контейнерные классы

Обычно **функционал классов-контейнеров** в C++ следующий:

- создание пустого контейнера (через конструктор);
- добавление нового объекта в контейнер;
- удаление объекта из контейнера;
- просмотр количества объектов, находящихся на данный момент в контейнере;
- очистка контейнера от всех объектов;
- доступ к сохраненным объектам;
- сортировка объектов/элементов (не всегда).

# C++STL

# STL – стандартная библиотека шаблонов

**Последовательные контейнеры**

std::vector;
std::deque;
std::array;
std::list;
std::forward_list;
std::basic_string.

**Ассоциативные контейнеры**

set
multiset
map
multimap

**Адаптеры**

stack
queue
priority_queue

# std::vector

Функции для работы с динамическими массивами **vector**

- size()
- resize()
- push_back()
- pop_back()
- clear()
- empty()

vector1 = push_back(1);

| 1 |
|---|

vector1 = push_back(4);

| 1 | 4 |
|---|---|

vector1 = push_back(8);

| 1 | 4 | 8 |
|---|---|---|

vector1 = and so on...

Each element is pushed at end of the vector

Consider this vector:

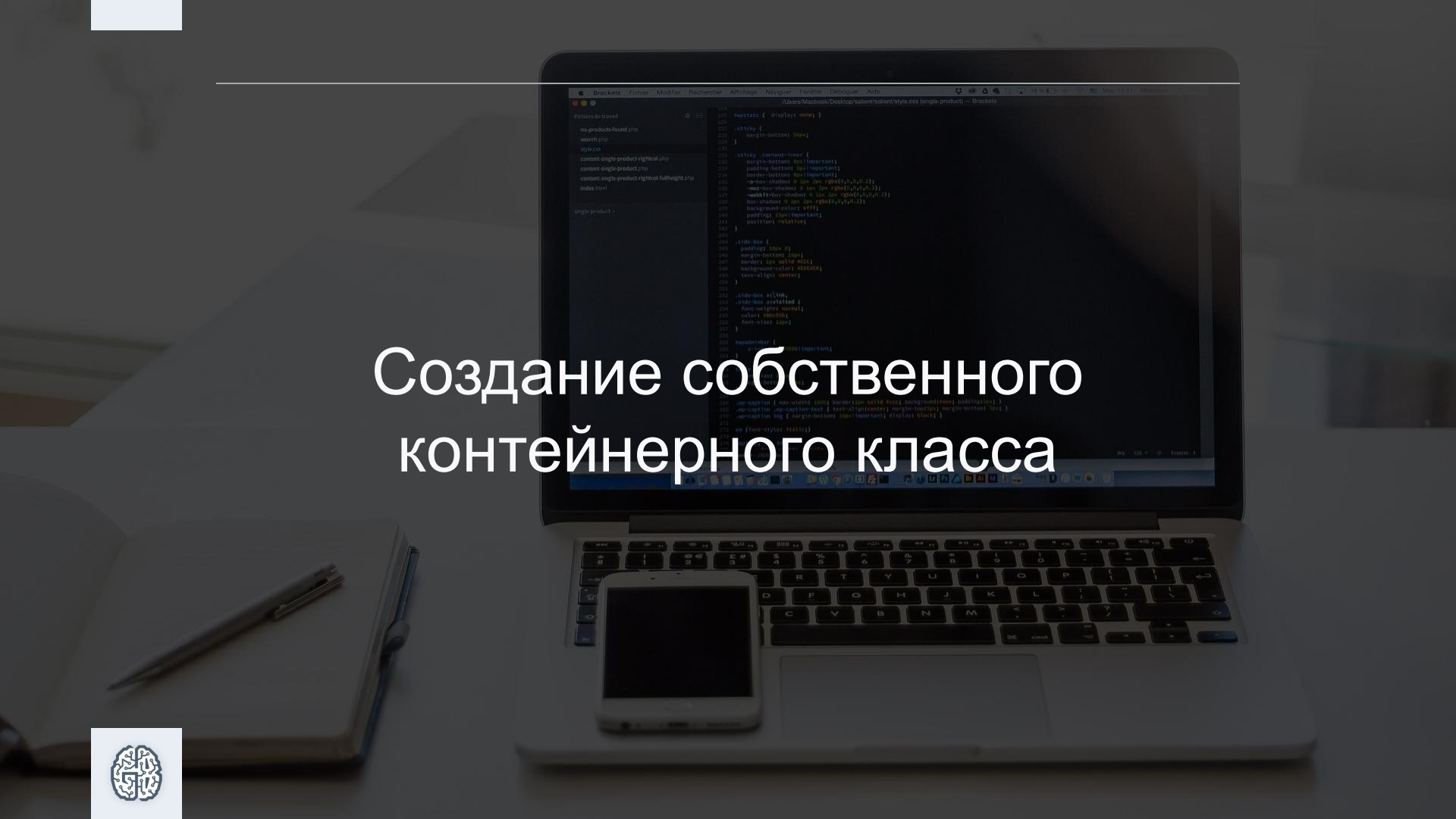
| 10 | 20 | 30 | 40 | 50 | 60 |
|---|---|---|---|---|---|

v =

after v.pop_back();

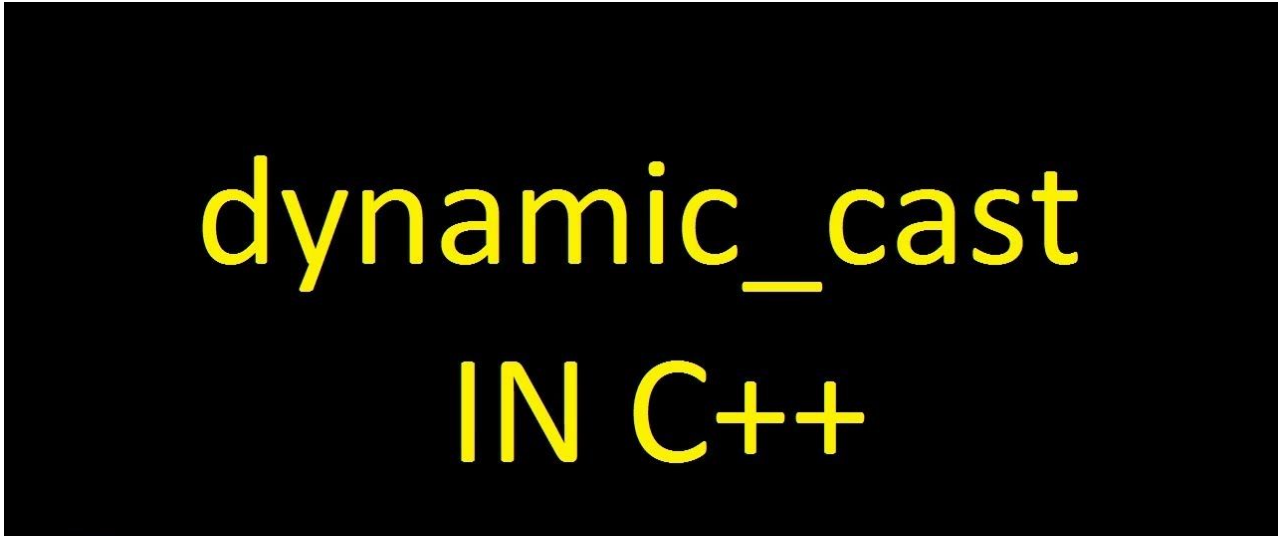
| 10 | 20 | 30 | 40 | 50 |
|---|---|---|---|---|

v =



# Создание собственного контейнерного класса

# Динамическое приведение типов




# Решите задачи

Какие типы отношений (композиция, агрегация, ассоциация или зависимость) описываются ниже?

1

Класс **Животное**, который содержит тип животного и его имя.

2

Класс **TextEditor** с методом **Save()**, который принимает объект **File**. Функция **Save()** записывает содержимое редактора на диск.

3

Класс **Авантюрист**, который может хранить разные **Предметы**: мечи, копья, зелья или книги заклинаний. Эти Предметы могут быть отброшены и подняты другими **Авантюристами**.

4

Программист использует **Компьютер** для просмотра видео с котами в интернете.

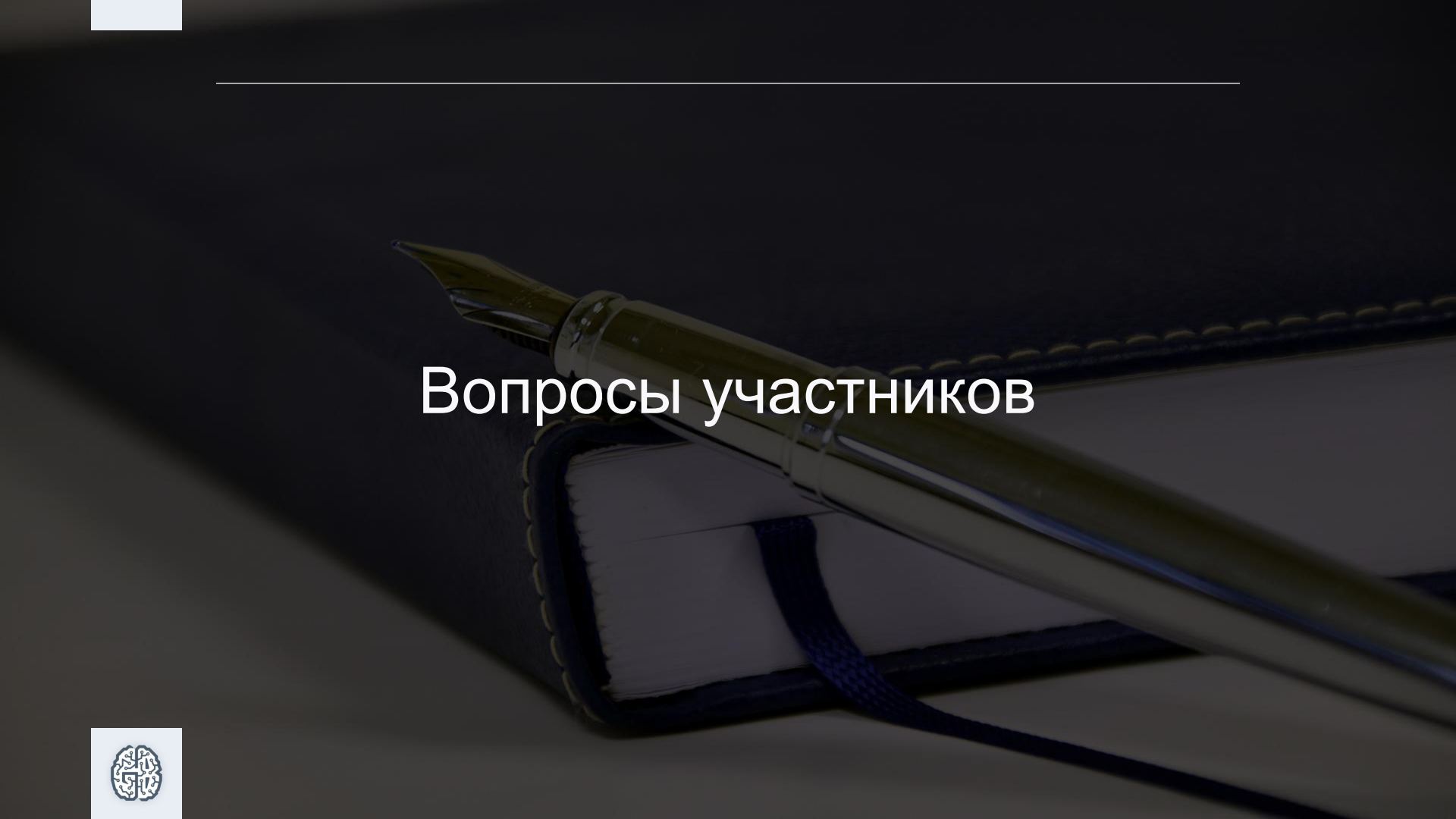
Найдите ошибку:

5

```
class B {
    virtual void f() { }
};

class V {
    virtual void g() { }
};

class X {
};

class D : public B, virtual public V,
virtual public X {
};
```

*Продолжение*

```
int main() {
    D d;

    X* px = &d;
    D* p1 = (D*)px;
    D* p2 = dynamic_cast<D*>(px);

    return 0;
}
```



# Вопросы участников