29. "Віртуальні конструктори" і клонування

Віртуальний конструктор це ідіома, яка дозволяє зробити те, підтримка чого в [C++](https://uk.wikibooks.org/wiki/C%2B%2B) не передбачена напряму. Ви можете добитися ефекту віртуального конструктора за допомогою віртуального метода класу virtual clone() (в якості конструктора копій) або методу virtual create() в якості конструктора за замовченням.

**class** **Shape** {

**public**:

**virtual** ~Shape() { } *// Віртуальний деструктор*

**virtual** void draw() = 0; *// Чисто віртуальна функція*

**virtual** void move() = 0;

...

**virtual** Shape\* clone() **const** = 0; *// Конструктор копій*

**virtual** Shape\* create() **const** = 0; *// Використовується як конструктор по замовченню*

};

**class** **Circle** : **public** Shape {

**public**:

Circle\* clone() **const**; *// Covariant Return Types; see below*

Circle\* create() **const**; *// Covariant Return Types; see below*

...

};

Circle\* Circle::clone() **const** { **return** **new** Circle(\***this**); }

Circle\* Circle::create() **const** { **return** **new** Circle(); }

У методі clone(), код new Circle(\*this) викликає конструктор копій класу Circle, щоб скопіювати this у новостворений об'єкт Circle. (Якщо клас Circle не є фінальним, ви можете зменшити ймовірність «розмазування» вказавши конструктору модифікатор доступу protected.) В методі create(), код new Circle() викликає конструктор за замовченням класу Circle's.

Інші програмісти користуються цими методами, так ніби вони є віртуальними конструкторами:

void userCode(Shape& s)

{

Shape\* s2 = s.clone();

Shape\* s3 = s.create();

...

**delete** s2; *// Тут необхідний віртуальний деструктор*

**delete** s3;

}

Ця функція працює коректно для всіх нащадків класу Shape, таких як: Circle, Square, або ще якихось нащадків, які можуть бути створені.

Зазначимо, що тип, який повертає функція clone() класу Circle навмисно відрізняється від типу, який повертає та сама функція класу Shape. Такі типи називається коваріантними вихідними типами - функціональність, яка спочатку не була частиною мови. Якщо ваш компілятор видає помилку при компіляції методу Circle\* clone() const в класі Circle (повідомляє, що значення що повертається відрізняється від визначення віртуальної функції базового класу), це означає що ви маєте старий компілятор і ви маєте змінити метод, так, щоб він повертав тип Shape\*.