

Лекція 7. Наслідування

Наслідування – механізм, який дозволяє одним об'єктам отримувати властивості інших об'єктів.

Наслідування має на увазі створення одного класу на основі іншого. Клас, на основі якого створюється новий клас, називається базовим. Новий клас, що створюється, називається похідним класом або підкласом.

Існує 3 типи наслідування: **public**, **private** та **protected**.

Синтаксис створення підкласу:



Типи наслідування

В залежності від типу поля базового класу (**public**, **private** та **protected**) та типу наслідування (**public**, **private** та **protected**).

<i>Механізм наслідування vs. тип члену</i>	public -наслідування	private -наслідування	protected -наслідування
public -член	public	private	protected
private -член	не наслідується	не наслідується	не наслідується
protected -член	protected	private	protected

Приклад: просте наслідування

```
class A{  
private:  
int x;  
public:  
int y;};
```

Базовий клас

```
class B: public A{  
public:  
int z;  
void show(){  
cout<<"y = "<<y<<endl;  
cout<<"z = "<<z<<endl;}};
```

Похідний клас

```
int main(){  
B obj;  
obj.y=1; obj.z=2;  
obj.show();  
return 0;}
```

Об'єкт похідного класу

Результат виконання

```
y = 1  
z = 2
```

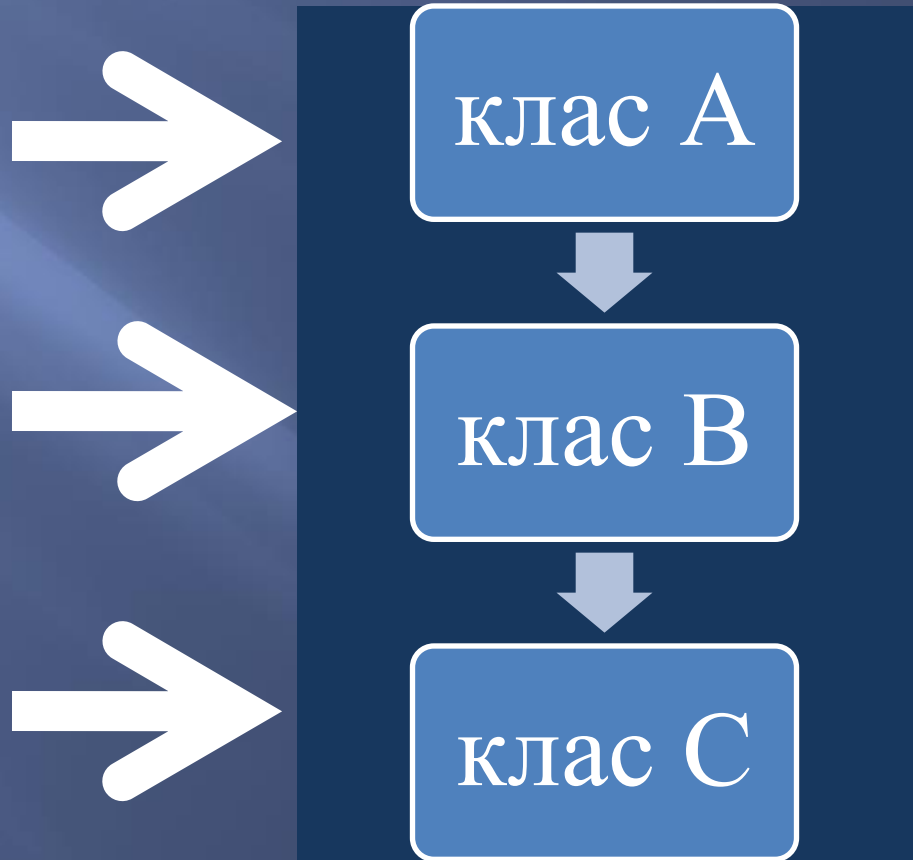
Багаторівневе наслідування

Похідний клас сам може бути базовим для іншого класу. В цьому випадку говорять про багаторівневе наслідування.

Клас А - базовий клас для класу В

Клас В – похідний від класу А і базовий для класу С

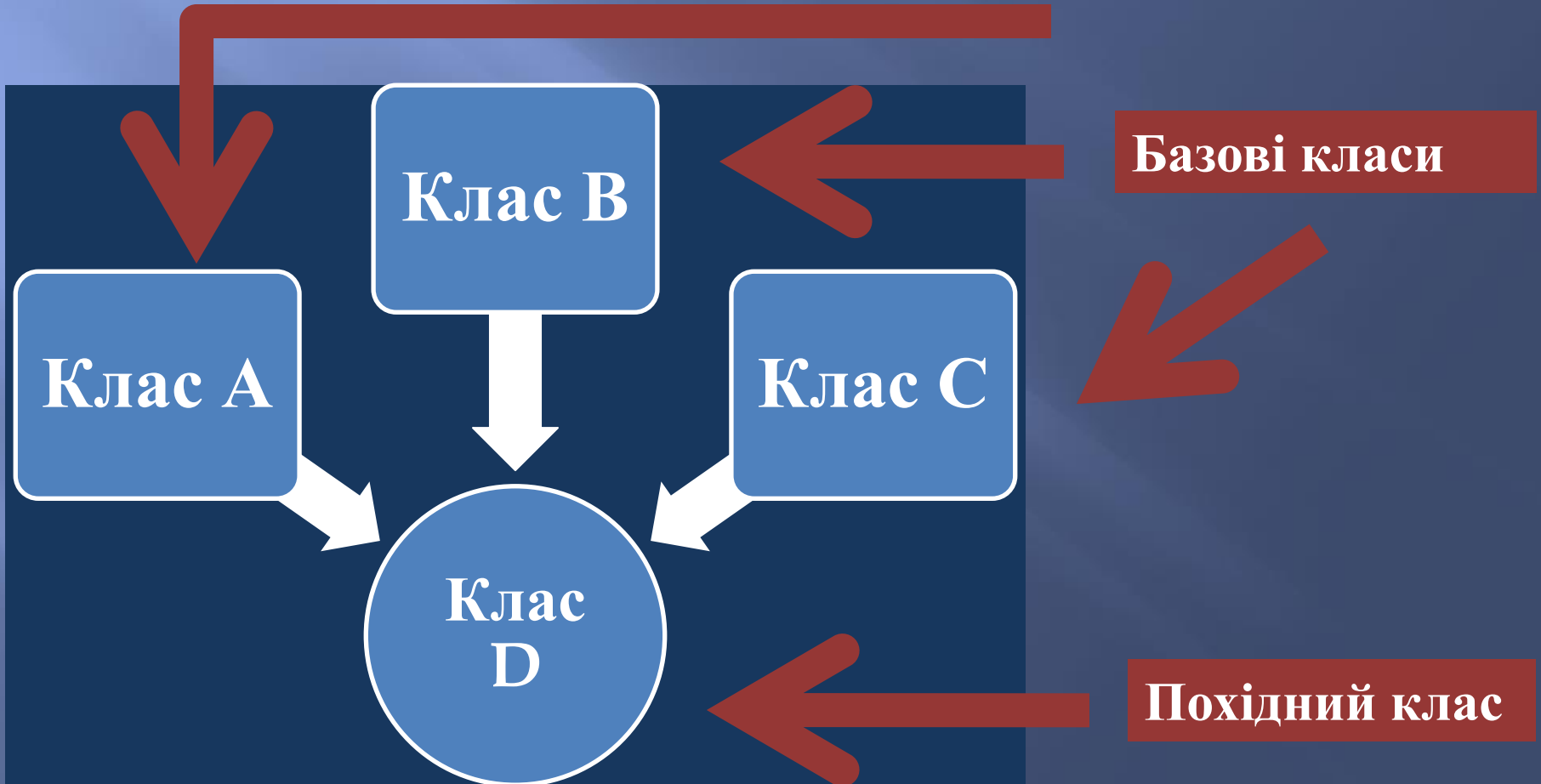
Клас С – похідний від класу В і може бути базовим для іншого класу



[Приклад \(pryklad01.cpp\)](#)

Багатократне наслідування

Похідний клас може створюватись на основі декількох базових класів. В цьому випадку говорять про багатократне наслідування. Базові класи в оголошенні похідного вказуються через кому разом з типом наслідування.



[Приклад \(pryklad02.cpp\)](#)

Переозначення методів і віртуальні функції

Методи базових класів в похідних класах можна переозначати. При цьому відповідний метод в базовому класі оголошують як віртуальний: в прототипі методу вказують ключове слово **virtual**

Віртуальність наслідується!

(якщо функція оголошена як віртуальна, то вона є віртуальною в усіх похідних підкласах)

Не слід плутати переозначення та перевантаження методів. При перевантаженні міняється прототип методу, але не міняється назва. При переозначенні не міняється прототип.

[Приклад \(przyklad03.cpp\)](#)

Конструктори і деструктори при наслідуванні

1. При створенні об'єкту похідного класу спочатку викликається конструктор базового класу.
2. Якщо базових класів декілька, їх конструктори викликаються у відповідності до списку базових класів в оголошенні похідного класу.
3. Деструктори викликаються в зворотному порядку.

При оголошенні **конструктора похідного класу** вказуються аргументи, що передаються конструкторам базових класів.

Синтаксис:

```
КПК(аргументи):КБК1(аргументи),КБК2(аргументи),...{
```

↑ // код
// конструктора похідного

//класу}

Конструктор похідного класу

[Приклад \(pryklad04.cpp\)](#)

Конструктори базових класів

Чисто віртуальні функції і абстрактні класи

Чисто віртуальною називається така віртуальна функція, що оголошена в базовому класі але не визначена в ньому.

Синтаксис:

```
virtual тип_рез-ту ім'я_функції(аргументи)=0;
```

Клас, який містить хоча б одну чисто віртуальну функцію, називається АБСТРАКТНИМ

Для абстрактного класу не існує об'єктів!



Зверніть увагу!

Ненаслідувані члени базового класу

Якщо член базового класу не наслідується в похідному класі, то це означає, що в похідному класі немає доступу до цього члена, однак технічно він існує!

[Приклад \(pryklad05.cpp\)](#)

РЕЗЮМЕ

1. **Наслідування** – механізм, що дозволяє одному об'єкту отримувати властивості іншого об'єкта.
2. Наслідування може бути **багатократним** та **багаторівневим**. При багатократному наслідуванні похідний клас створюється на основі декількох базових класів. При багаторівневому наслідуванні похідний клас є базовим для іншого класу.
3. Методи базового класу можна **переозначати**. Метод, що переозначається в похідному класі, в базовому оголошується як **віртуальний** (ключове слово **virtual**).
4. При створенні об'єкта похідного класу спочатку викликаються **конструктори** базових класів (зліва направо в списку наслідування). **Деструктори** викликаються в зворотному порядку.