ROBOCODE
SENIOR EMBEDDED

ООП. Класи в C++

LESSON 26. FOR STUDENTS

English Time

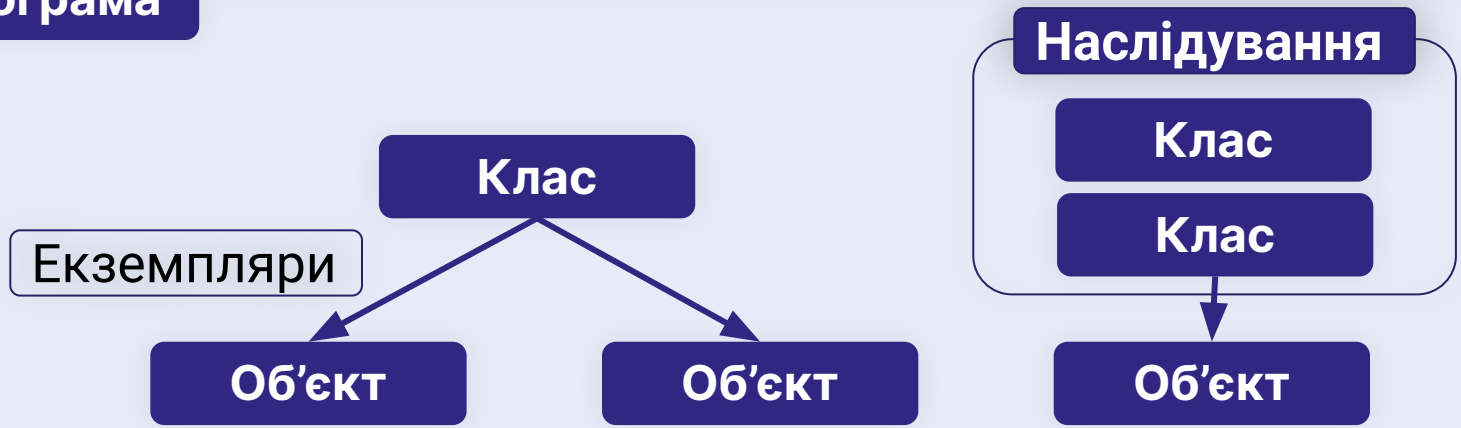
Час Англійської

Method	Метод
Constructor	Конструктор
Encapsulation	Інкапсуляція
Field	Поле

ООП

ООП - Об'єктно-орієнтоване програмування - методологія програмування, заснована на представленні програм у вигляді сукупності об'єктів

Програма

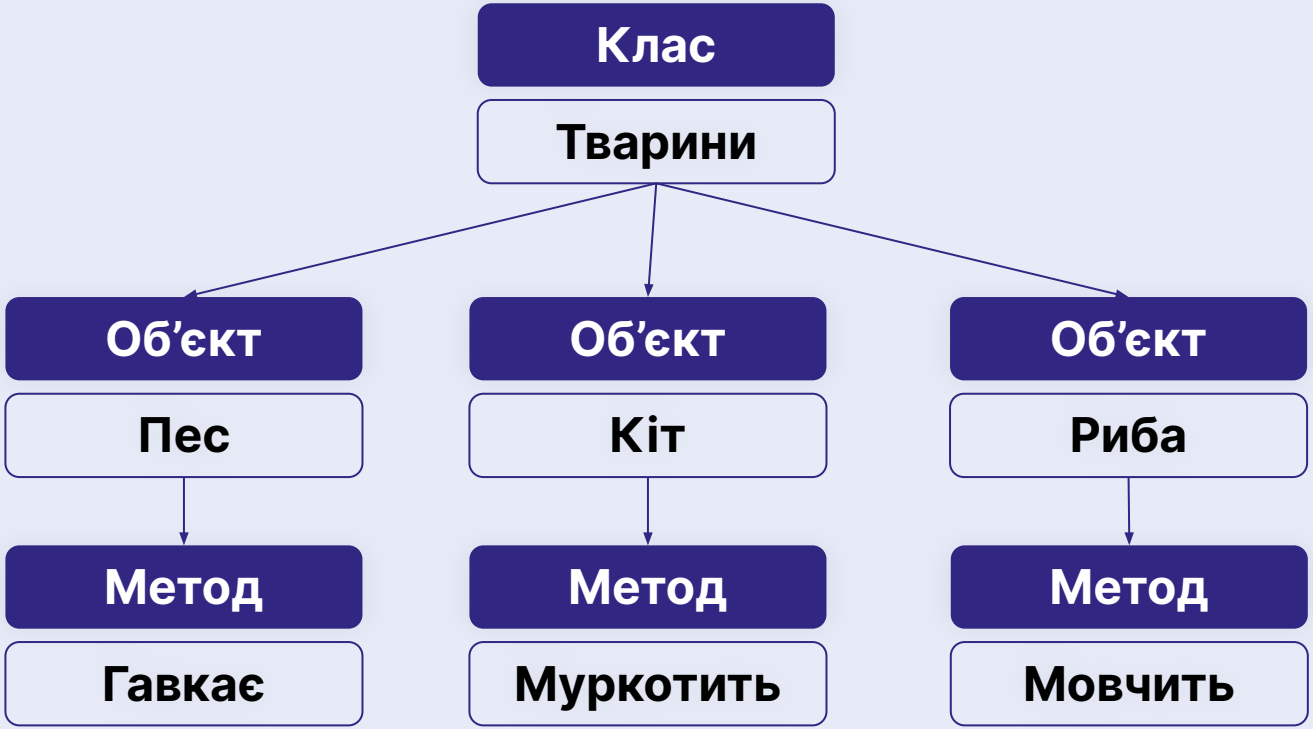


Кожен з яких є екземпляром визначеного класу

Класи формують ієрархію наслідування

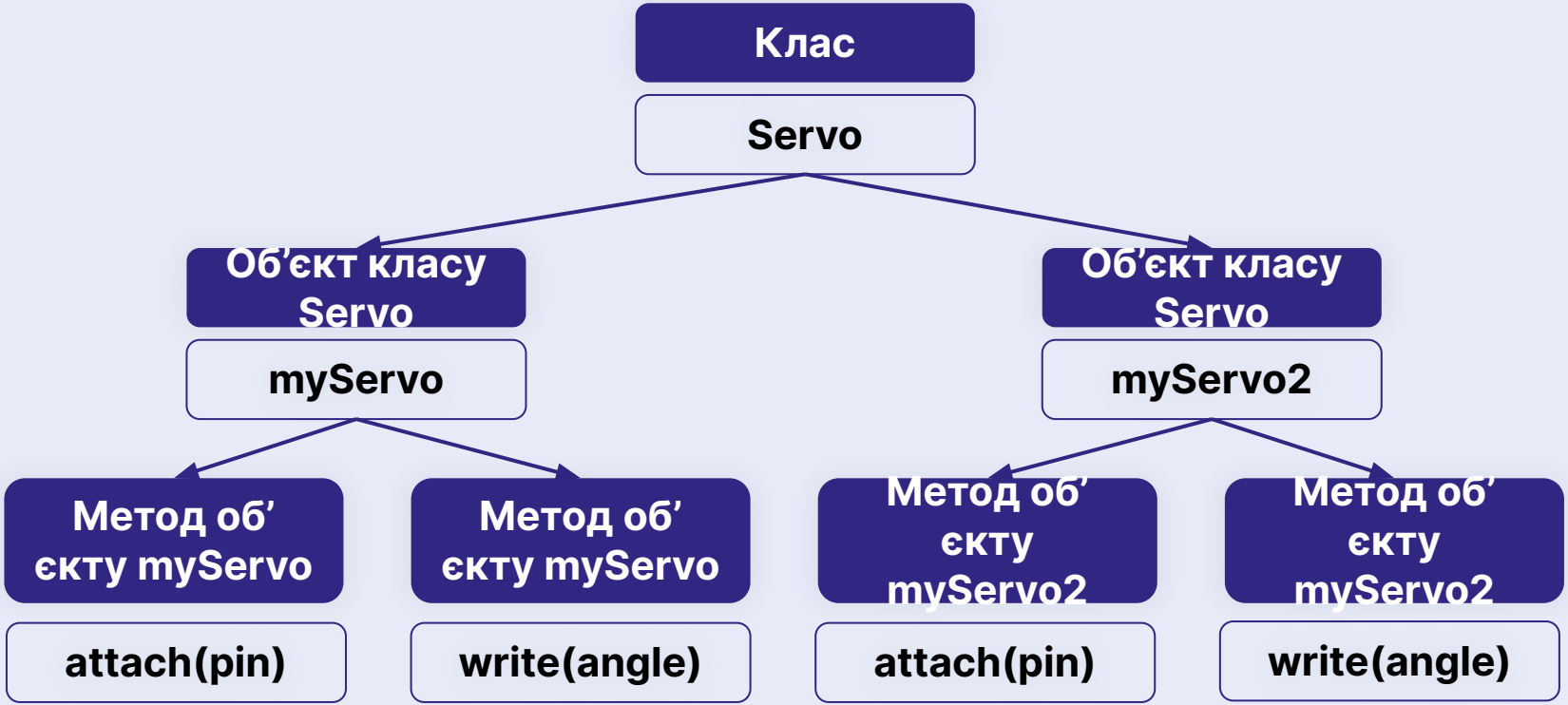
ООП

Метод в ООП - це функція, що належить якомусь об’єкту. Один об’єкт може мати необмежену кількість методів



ООП

Розглянемо структуру на прикладі класу **Servo**



ООП

Розглянемо приклад в вигляді коду

```
#include<Servo.h>
Servo myServo;
Servo myServo2;
void setup() {
  myServo.attach(8);
  myServo2.attach(9);
}
void loop() {
  myServo.write(180);
  myServo2.write(90);
}
```

Клас

Клас - конструкція, що групує набір змінних та функцій, для реалізації схожих завдань

Створимо клас **собака**

```
class Dog{
```

Задамо загальні властивості всіх собак:

- колір
- вага
- хвіст

```
String color;
int mass;
float tail;
```

Задамо загальні дії всіх собак:

- гавкає
- спить

```
void gav(){
  Serial.print("gav");
}
void sleep(int time){
  Serial.print("hrrrrr");
}
```

Клас собака

Повний код для створеного класу **собака**

```
class Dog{
public:
  String color;
  int mass;
  float tail;

  void gav(){
    Serial.print("gav");
  }
  void sleep(int time){
    Serial.print("hrrrrr" + time);
  }
}
```

Клас собака

- У нас є шаблон, що матиме собака, а тепер давайте ми її створимо і назвемо Рекс
- Наш Рекс буде білого кольору, матиме вагу 10кг та хвіст довжиною 15см.
- Ми попросимо Рекса гавкнути і піти спати 120хв

```
Dog Rex;
void setup(){
  Rex.color="white";
  Rex.mass = 5;
  Rex.tail = 20.5;
}
 void loop(){
  Rex.gav();
  Rex.sleep(120);
 }
```

Поле класу

Властивості нашого собаки називаються **полями класу**

```
String color;
int mass;
float tail;
```

Дії, що може виконувати цей собака називаються **методами класу**

```
void gav(){
  Serial.print("gav");
}
void sleep(int time){
  Serial.print("hrrrrr"+time);
}
```

Ім’я, що ми надали нашому собаці при створенні - це **екземпляр класу**

```
Dog Rex;
```

Клас риби

- Написати власний клас для риби та створити об’єкт
- Клас матиме назву **Fish**
- Клас матиме три параметри:
  - вага (кг)
  - хижа або не хижа
  - місце проживання
- Клас містить функцію їсти (кг) та плавати (швидкість)
- Створити об’єкт класу з ім’ям **shark.** Та задати власні параметри і викликати методи цього класу

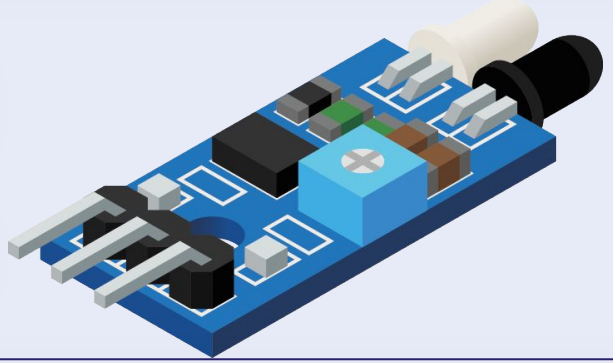
Реалізація коду

```
class Fish {
  public:
    int mass;
    bool predator;
    String location;
    void eat(int kg) {

    }
    void swim(int speed) {

    }
};

Fish shark;
void setup() {
  shark.mass = 1100;
  shark.predator = true;
  shark.location = "Ocean";
  shark.eat(30);
  shark.swim(40);
}

void loop() {
}
```

i Вітаю, ми створили шаблон білої акули

IR Sensor

- Що це таке?
- Як працює цей датчик?
- Який сигнал він передає?



- Написати програму для зчитування даних із датчику
- Створити змінну для визначення порту. Створити **функцію** sensorRead, що буде повертати це значення
- В циклі, вивести значення в монітор порту

Реалізація коду

```
int sensPin=9;
void setup() {
  Serial.begin(9600);
  pinMode(sensPin,INPUT);
}

int sensorRead(){
  return digitalRead(sensPin);
}

void loop() {
   Serial.println(sensorRead());
}
```

Практичне завдання

- Щоб запустити датчик, необхідно визначити *номер порту* та *режим порту*, а також викликати *відповідну функцію* для зчитування даних
- Якщо наш проект містить 10 таких датчиків, то ці операції необхідно повторювати 10 разів.
- Ми спробуємо створити для програми один узагальнений клас IRSensor, а далі для кожного датчику будуть власні параметри
- Створити клас **IRSensor** із полем **sensPin** і методом **sensorRead**, що буде налаштовувати порт і повертатиме значення датчику

Реалізація коду

```
class IRSensor {
  public:
    int sensPin;
    int sensorRead() {
      pinMode(sensPin,INPUT);
      return digitalRead(sensPin);
    }
};

IRSensor ir;
void setup() {
 Serial.begin(9600);
 ir.sensPin=9;
}

void loop() {
Serial.println(ir.sensorRead());
}
```