# Наследование

# Синтаксис объявления производного класса

class *имя\_производного\_класса* :

тип\_наследования *имя\_базового\_класса*

{

};

class Animal { … };

class Dog : public Animal { … } ;

# Спецификатор (тип) наследования

class Dog : <**тип\_наследования>** Animal

{

…

};

|  |  |  |
| --- | --- | --- |
| Спецификатор доступа в базовом классе | Доступ к данным базового класса | |
| с помощью объекта производного класса | из методов производного класса |
| private | нет доступа | нет доступа |
| protected | нет доступа | есть доступ |
| public | есть доступ | есть доступ |

# Передача аргументов в конструктор базового класса

class Animal

{

int m\_age;

char\* m\_pName;

public:

Animal(int age, char\* pName);

};

Dog.h

class Dog : public Animal {

bool m\_hasMaster;

public:

Dog(int age, char\* pName, bool hasMaster);

};

Dog.cpp

Dog::Dog(int age, char\* pName, bool hasMaster)

**:Animal(age, pName)**

{

m\_hasMaster = hasMaster;

}

# Передача параметров конструктору копирования базового класса

Dog.h

class Dog : public Animal {

bool m\_hasMaster;

public:

Dog( const Dog& other );

…

};

Dog.cpp

Dog::Dog ( const Dog& other )

:Animal ( other )

{

m\_hasMaster = hasMaster;

}

# Оператор присваивания и наследование

Dog.h

bool operator = (const Dog & other);

Dog.cpp

Dog & Dog :operator = (const Dog & other)

{

if (this != & other)

{

Animal::operator=( other);

//копирование базовой части

…

}

return \*this;

}

# Вызов сокрытых методов базового класса

class Animal

{

public:

void makeNoise() const { cout << "I am an animal"; }

void makeNoise(int r) const { cout << "I am a new animal"; }

};

class Dog : public Animal

{

public:

void makeNoise() const { cout << “Woof!!!"; }

};

int main()

{

Dog dog;

dog. makeNoise(); //Вызов метода класса Dog

dog**::Animal**.makeNoise(); //Вызов метода класса **Animal**

dog**::Animal**.makeNoise(10); //Вызов метода класса **Animal**

Dog\* dog1 = new Dog();

dog**::Animal** -> makeNoise(10); //Вызов метода класса **Animal** c помощью указателя

}