

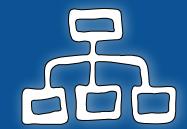
Алгоритмы и Алгоритмические Языки (С++)

Семинар #8:

- 1. Наследование в С++: применение и реализация.
- 2. Tunы наследования: private, protected, public.
- 3. Виртуальные функции: применение и реализация.
- 4. Иерархия классов в примитивной игре.



Наследование в С++: применение и реализация.







```
Object
protected:
    Pos x_, y_;
    Pos sz_x_, sz_y_;
    Vel v_x_, v_y_;
public:
    Object(Pos x, Pos y, ...);
    void move(Time_t dt);
    void accelerate(Acc a_x, Acc a_y, Time_t dt);
    void dissipate(double koeff, Time_t dt);
    void wall_bounce(Pos x0, Pos y0, ...);
                            public
                    MassiveObject
private:
    Mass mass_;
public:
    MassiveObject(Pos x, Pos y, ..., Mass mass);
    void apply_force(Force f_x, Force f_y, ...);
```



Tunы наследования: private, protected, public







```
Base
private:
    int base_private_;
protected:
    int base_protected_;
public:
    int base_public_;
                             public
                       Derived
private:
protected:
    int base_protected_;
public:
    int base_public_;
```





Base	
<pre>private: int base_private_;</pre>	
<pre>protected: int base_protected_;</pre>	
<pre>public: int base_public_;</pre>	
	protected
Derived	
private:	
<pre>protected: int base_protected_; int base_public_;</pre>	
public:	





Base	
<pre>private: int base_private_;</pre>	
<pre>protected: int base_protected_;</pre>	
<pre>public: int base_public_;</pre>	
private	
Derived	
<pre>private: int base_protected_; int base_public_;</pre>	
protected:	
public:	



Виртуальные функции: применение и реализация





Иерархия классов в примитивной игре







```
Object
protected:
    Pos x_, y_;
    Pos sz_x_, sz_y_;
    Vel v x , v v ;
public:
    Object(Pos x, Pos y, ...);
    void move(Time_t dt);
    void accelerate(Acc a_x, Acc a_y, Time_t dt);
    void dissipate(double koeff, Time_t dt);
    void wall_bounce(Pos x0, Pos y0, ...);
                    MassiveObject
private:
    Mass mass ;
public:
    MassiveObject(Pos x, Pos y, ..., Mass mass);
    void apply_force(Force f_x, Force f_y, ...);
```

```
RenderedObject
public:
    RenderedObject(Pos x, Pos y, ..., Mass mass);
    virtual void render(Renderer& renderer);
    virtual ~RenderedObject();
                        Round
public:
    Round(Pos x, Pos y, Pos diameter, ...);
    virtual void render(Renderer& renderer);
    virtual ~Round();
                       Square
public:
    Square(Pos x, Pos y, Pos sz_x, Pos sz_y, ...);
    virtual void render(Renderer& renderer);
    virtual ~Square();
```

Вопросы?

