

# Принципы ООП



# Принципы

- Абстракция
- Инкапсуляция
- Наследование
- Полиморфизм

# Абстракция

```
1 class Human {  
2     public:  
3         virtual void say(void) = 0;  
4 };  
5  
6 class Petya : public Human {  
7     public:  
8         std::string feature = "лысеть в 30";  
9         void say() {std::cout << "Hi! I' m Petya!!!" << std::endl;}  
10 };  
11
```

//Hi! I' m Petya!!!

# Инкапсуляция

```
14 class Account {
15     private:
16         std::string login = "Petya1976";
17         std::string password = "qwerty1976";
18     public:
19         std::string name = "Пётр";
20         int birth_year = 1976;
21 };
22
```

```
30 int main()
31 {
32     Account Petr76;
33     std::cout << Petr76.login << std::endl;
34     return 0;
35 }
```

//Ошибка компиляции

```
30 int main()
31 {
32     Account Petr76;
33     std::cout << Petr76.name << Petr76.birth_year << std::endl;
34     return 0;
35 }
```

//Пётр76

# Наследование

```
6 class Petya : public Human {  
7     public:  
8         std::string feature = "лысеть в 30";  
9         void say() {std::cout << "Hi! I'm Petya!!!" << std::endl;}  
10};
```

```
23 class Ivan_Petrovich : public Petya {};
```

```
29  
30 int main()  
31 {  
32     Ivan_Petrovich Vanya;  
33     std::cout << Vanya.feature << std::endl;  
34     return 0;  
35 }  
36
```

//Лысеть в 30

# Полиморфизм

```
6 class Petya : public Human {  
7     public:  
8         std::string feature = "лысеть в 30";  
9         void say() {std::cout << "Hi! I'm Petya!!!" << std::endl;}  
10};
```

```
23 class Ivan_Petrovich : public Petya {};
```

```
24
```

```
25 class Petr_Petrovich : public Petya {};
```

```
30 int main()  
31 {  
32     Ivan_Petrovich Vanya;  
33     std::cout << Vanya.feature << std::endl;  
34  
35     Petr_Petrovich Petro;  
36     std::cout << Petro.feature << std::endl;  
37     return 0;
```

//Лысеть в 30

//Лысеть в 30

Спасибо за внимание

*ссылка на примеры классов:  
<https://pastebin.com/TNVaQKYQ>*

