

1. Предположим, что при разработке игры вам потребовался юнит "Челмедведосвин" (*ManBearPig*).

Разработайте иерархию классов, реализуйте необходимые конструкторы и добавьте разумные методы.

PS: В данном игровом контексте предполагается, что человек — не животное.

Структуры, которые обязаны присутствовать: Unit, Animal, Man, Bear, Pig

2. При помощи typedef определите тип ComplexFunction, который является указателем на функцию, которая принимает int и указатель на функцию, принимающую double и возвращающую int, и возвращает указатель на функцию, которая принимает char const * и возвращает int *.
3. Напишите возвращающую bool шаблонную функцию compare, которая принимает две константные ссылки на объекты одного типа и указатель на константный метод этого типа без параметров, который в свою очередь возвращает значение какого-то второго типа. Функция должна сравнивать объекты по значениям, которые для них вернёт соответствующий метод, и возвращать true, если значение для первого объекта оказалось меньше, чем для второго.

```
std::string s1("Elf");
std::string s2("Archer");

// сравнение строк по длине
bool r1 = compare(s1, s2, &std::string::size); // true
bool r2 = compare(s1, s1, &std::string::size); // false
```

4. Предположим, что вам даётся два указателя на объект некоторого сложного полиморфного класса, который использует множественное наследование. Оба указателя имеют один и тот же тип, но могут ссылаться на разные однотипные базы внутри одного объекта.

Напишите шаблонную функцию, которая по двум однотипным указателям проверяет, указывают ли они на один объект.

```
template<class T>
bool isSameObject(T const * p, T const * q){
```