Задача 1.

```
#include <iostream>
#include <vector>
class Animal {
public:
    virtual void say() {}
};
class Wolf : public Animal {
public:
    virtual void say() {
        std::cout << "Woooo!" << std::endl;
    }
};
class Raccoon : public Animal {
public:
    virtual void say()  {
        std::cout << "Trill trill!" << std::endl;
    }
};
class Duck : public Animal {
public:
    virtual void say()  {
        std::cout << "Quack quack!" << std::endl;
    }
};
class Dolphin : public Animal {
public:
    virtual void say()  {
        std::cout << "Click click!" << std::endl;
    }
};
class Moose : public Animal {
public:
    virtual void say()  {
        std::cout << "Bellow bellow!" << std::endl;
    }
};
int main() {
    std::vector<Animal*> zoo;
    zoo.push_back(new Duck());
    zoo.push_back(new Dolphin());
    zoo.push_back(new Moose());
    zoo.push_back(new Raccoon());
    zoo.push_back(new Wolf());
    for(Animal * animal: zoo) {
        animal->say();
    }
    return 0;
}
```

Задача 2.

```cpp
#include <iostream>
#include <cmath>

class Callback {
public:
    virtual void exec(double data) {}
};

class Computator {
public:
    void compute(double data, Callback * callback) {
        double result;
        result = exp(tan(fabs(sin(data))));
        callback->exec(result);
    }
};

class Printer : public Callback {
public:
    virtual void exec(double data)  {
        std::cout << data << std::endl;
    }
};

class DoubledCompute : public Callback {
public:
    DoubledCompute(Computator * c) {
        computator = c;
    }
    virtual void exec(double data)  {
        Callback * pr = new Printer();
        computator->compute(data, pr);
        delete pr;
    }
private:
    Computator * computator;
};

int main() {
    Computator * computator = new Computator();
    Callback * printer = new Printer();
    Callback * doubled = new DoubledCompute(computator);
    double num;
    std::cin >> num;
    computator->compute(num, printer);
    computator->compute(num, doubled);
    return 0;
}
```