

**Классная работа 2**  
**Классы. Статические, константные методы. Динамическая память.**  
**Наследование. Виртуальные функции.**  
**26.11.2018**

1. Описать 2 класса: товар (`class Good`) и каталог (`class Catalog`). Товар описывается 4 параметрами: порядковый номер (`const int id`), название (`const string name`), цена (`const int price`) и количество товара на складе (`int number`). Каталог содержит количество различных товаров (`int Catalog::numberOfGoods`), которое передается в конструкторе, и сам массив указателей на товары (`class Catalog::Goods ** list`), который создается динамически в конструкторе (а очищается в деструкторе). Товары создаются тоже динамически. Порядковые номера товаров присваивает каталог (начиная с 0). Необходимо описать методы покупки (`void Catalog::buy(string name, int number)` и `void Catalog::buy(int id, int number)`), который проверяет, возможна ли покупка данного товара в указанном количестве (если покупка осуществилась успешно, то вычесть указанное количество товара со склада). После обработки всех запросов на покупки распечатать весь каталог (`void Catalog::print() const`).

На вводе дано число  $n$  — количество товаров в каталоге. Далее каталог формируется  $n$  запросами вида `+ name price number`. Потом следуют число  $m$  — количество запросов на покупку. Далее запросы на покупку вида `? name` или `# id`.

Замечание: класс каталог является другом к классу товар.

Ввод	<pre> 3 + water 100 33 + juice 200 21 + banana 150 29 5 ? banana 10 # 0 15 # 2 40 # 5 10 ? banan 10 </pre>
Вывод	<pre> banana x 10 = 1500 water x 10 = 1000 banana x 40 = not enough 5 = incorrect index banan = incorrect name # name price number 0 water 100 23 1 juice 200 21 2 banana 150 19 </pre>

2. Описать 2 класса: герой (`class Hero`) и оружие (`class Weapon`). Герой описывается 4 параметрами: имя (`string`), атака (`int attack`), защита (`int protection`) и оружие (`Weapon weap`). Оружие описывается 2 параметрами: тип оружия (внутренний перечислимый тип `enum type{ATTACK, PROTECTION, NONE}`) и сила оружия (`int value`). Атакующая сила героя — это сумма атаки героя и силы атакующего оружия (если оружие атакующего типа). Защитная сила героя — это сумма защиты героя и силы защитного оружия (если такое оружие есть). Герой *A* побеждает героя *B* если, атакующая сила героя *A* строго больше защитной силы героя *B*.

Дано целое  $n$  — количество героев и количество единиц оружия. Далее вводится  $n$  имен героев (изначально ни у кого оружия нет, то есть тип `NONE`), их нападение и защиту. Потом следует целое  $m$  — количество единиц оружия. Далее вводится  $m$  строк, описывающих оружие (тип оружия, сила оружия и имя героя). Если у героя уже есть оружие и ему предлагают еще одно, то он выбирает оружие с большей силой. Для сравнения оружия опишите метод `bool Weapon::operator>(const Weapon &)`. Для сравнения героев опишите метод `bool Hero::operator>(const Hero&)`. Вывести кто кого побеждает (перебрать все пары различных героев).

Замечание: нельзя использовать дружественные классы и методы (при необходимости описать методы `get() const`), массив героев сделать динамическим массивом в `int main()`.

Ввод	3 Dima 5 5 Kuat 6 1 Pasha 4 4 5 1 attack Pasha 4 protect Kuat 3 protect Pasha 2 attack Kuat 5 attack Pasha
Вывод	Dima win Pasha Kuat win Dima Kuat win Pasha Pasha win Dima Pasha win Kuat

Пояснение к примеру: Дима без оружия (атака: 5, защита: 5), Куат с защитным оружием силы 4 (атака: 6, защита: 5), Паша с атакующим оружием силы 5 (атака: 9, защита: 4).

3. Описать класс `Pair`, содержащий 2 поля и конструкторы. От него отнаследовать 2 класса: рациональные числа и комплексные числа (с целыми значениями по отдельным компонентам). Определить операции сложения, вычитания и умножения и производных классов. Изначально имеется 1 рациональное число  $0/1$  и одно комплексное  $0+1*i$ . Необходимо провести с ними некоторые операции. У класса рациональных чисел определить функцию сокращения (делить числитель и знаменатель на НОД, который ищется алгоритмом Евклида). У класса `Pair` и наследников определить виртуальный метод `print()`.

На вводе дано число  $n$  — количество запросов. Далее  $n$  запросов, которые преобразуют числа (сначала идет тип операции: 1 или 2; потом операция:  $+$ ,  $-$ ,  $*$ ; потом число, определенное 2 числами).

Ввод	6 1 + 2 3 1 + 1 3 1 - 1 3 1 * 1 2 2 + 1 0 2 * 0 1
Вывод	2/3 1/1 2/3 1/3 1+1*i -1+1*i

4. Описать классы: четырехугольник `Quad`, параллелограмм (отнаследован от четырехугольника), квадрат (отнаследован от параллелограмма). Класс четырехугольник имеет поля: 4 стороны и виртуальные методы `void print()`.

Дано  $n$ , далее  $n$  описаний сторон (4 числа). По вводимым данным определять тип четырехугольника (для этого определить статические методы `bool is(int, int, int, int)`) и создавать соответствующий объект динамически в массив указателей `Quad **ptr`. Далее распечатать все объекты (тип объекта печатается в соответствующих методах `void print()`).

Ввод	5 1 2 3 4 1 2 2 1 1 2 1 2 2 2 2 2 3 2 3 2
Вывод	Quad: 1 2 3 4 Quad: 1 2 2 1 Parallelogram: 1 2 Square: 2 Parallelogram: 3 2

5. Описать класс `class Computer` с параметрами частоты процессора `int clockRate` и объемом оперативной памяти `int RAM` и соответствующими `get`-функциями. Отнаследовать класс `class Notebook` с параметрами: длительность батареи `int Batery` и название производителя `string Vendor`. Отнаследовать класс `class Monoblock` с параметром: диагональ `int Display`.

На вводе дано число  $n$  — количество аппаратов. Далее  $n$  описаний устройств:

? — неизвестное устройство,

N — ноутбук,

M — моноблок.

Распечатать устройства с максимальной частотой и максимальным объемом оперативной памяти.

Ввод	6 N 2000 8000 5 ASUS ? 3000 4000 M 2500 6000 25 M 1500 2000 21 N 2500 6000 7 HP ? 2300 8000
Вывод	Maximum clockRate: Monoblock 2500 GHz 6000 Mb 25" Notebook 2500 Ghz 6000 Mb 7 hour HP Maximum RAM: Notebook 2000 Ghz 8000 Mb 5 hour ASUS Computer 2300 Ghz 8000 Mb