

### Вопрос 1

Укажите **главные** концепции объектно-ориентированного подхода

модульность

параллелизм

инкапсуляция (ограничение доступа)

типизация

абстрагирование

иерархия

сохраняемость

### Вопрос 2

```
void f()
{
    ... // 1
    {
        ... // 2
        int i;
        for(i=0; i<10; ++i)
        {
            ... // 3
        }
        ... // 4
    }
    ... // 5
}
```

Укажите все области действия переменной i

3

5

4

2

1

### Вопрос 3

Какое из следующих объявлений создает константный указатель на изменяемый символ?

char\* const p;

const char\* p;

&const\* char p;

const char& p;

нет правильного варианта

### Вопрос 4

Укажите правильный вариант для перегрузки операции delete в классе A:

```
class A {
public:
    ...
    void operator delete(A *);
    void delete(void);
    void operator delete(void *p, size_t size);
    static void operator delete(size_t size, A *);
}
```

## Вопрос 5

Укажите правильный заголовок для перегрузки операции вывода значения класса X:

```
void operator <<(ostream &, const X);
ostream &operator <<(X &, ostream &);
friend ostream operator <<(ostream, X);
ostream &operator <<(ostream &, const X &);
void operator <<(ostream &, X &);
```

## Вопрос 6

Чтобы объявить константу MAXSIZE в классе A перед ней нужно написать

```
class A {
    _____ int MAXSIZE=100;
    ...
};
```

## Вопрос 7

Деструктор - это метод класса, который  
 не может вызывать другие методы этого класса  
 может быть автоматически создан компилятором, если не определен в классе  
 должен быть без аргументов и возвращаемого значения  
 должен всегда вызываться явно  
 должен иметь имя из символа ~ и имени класса  
 может иметь произвольное имя, но начинающееся с символа ~  
 не может быть виртуальным

## Вопрос 8

```
void f(int &); // 1
void f(double); // 2
void f(int *); // 3
```

```
f(10);
```

В последней строке примера будет  
вызвана функция 2  
ошибка компиляции  
вызвана функция 3  
вызвана функция 1

Вопрос 9

Нельзя перегружать следующие операции (укажите все правильные варианты):

•  
::  
&&  
,  
->

Вопрос 10

Если для хранения данных объекта некоторого класса динамически выделяется память, то для этого класса необходимо определить:

конструктор по умолчанию  
конструктор копий  
операцию new  
конструктор-преобразователь  
операцию присваивания  
деструктор

Вопрос 11

```
struct A {  
public:  
    void f();  
};  
struct B : A {  
public:  
    void g();  
};
```

Метод  $f$  класса  $A$  в производном классе  $B$  является

public  
protected  
private

Вопрос 12

```
class A { public: A(int n); };
```

```
class B { public: B(int n); };
class C { public: C(int n); };
class D : B, A {
    C c1;
public:
    D(int n) : A(n), B(0), c1(1) {}
};
```

Укажите порядок выполнения конструкторов при выполнении конструктора D:

A, B, C

C, B, A

B, A, C

C, A, B

### Вопрос 13

```
class A {
    int n;
public:
    A() { n=0; }
    A(int k) { n=k; }
};
```

Укажите все правильные варианты порождения исключительной ситуации класса A:

throw A(int k=10);

throw new A(10);

throw A;

throw A(10);

throw A();

throw 10;

### Вопрос 14

```
double a=52.74;
```

```
int b;
```

Для преобразования вещественного числа a в целое нужно использовать оператор:

b=static\_cast<int>(a);

b=reinterpret\_cast<int>(a);

b=dynamic\_cast<int>(a);

b=const\_cast<int>(a);

это сделать невозможно

### Вопрос 15

```
template void f<int *>(int *a);
```

Это объявление является  
явной специализацией шаблона функции f  
частичной специализацией шаблона функции f  
определением шаблона функции f  
неявным инстанцированием шаблона функции f  
явным инстанцированием шаблона функции f

## Вопрос 16

Среди алгоритмов в STL есть  
меньшее из двух значений  
решение системы линейных уравнений  
решение квадратного уравнения  
замена элементов последовательности с заданным значением  
слияние двух отсортированных последовательностей

## Вопрос 17

Укажите соответствие между паттернами и их назначением:

1. Мост
2. Адаптер
3. Фасад
4. Заместитель

Является суррогатом другого объекта и контролирует доступ к нему  
Отделить абстракцию от ее реализации так, чтобы то и другое можно было изменять независимо  
Преобразует интерфейс одного класса в интерфейс другого, который ожидают клиенты  
Предоставляет унифицированный интерфейс вместо набора интерфейсов некоторой подсистемы

## Вопрос 18

Укажите правильно спроектированный класс:

```
class Date {  
    int day, month, year;  
public:  
    Date(int d, int m, int y);  
    int getDay() const;  
    int getMonth() const;  
    int getYear() const;  
    void setDate(int d, int m, int y);  
    ...  
};
```

```

class Date {
    int day, month, year;
public:
    Date(int d, int m, int y);
    int &Day() { return day; }
    int &Month() { return month; }
    int &Year() { return year; }
    ...
};
class Date {
public:
    int day, month, year;
    Date(int d, int m, int y);
    ...
};

```

### Задача

Определить операцию + для сложения двух значений класса матрица:

```

class Matrix
{ double **data; // массив чисел
  int size1,size2; // размеры
public:
    Matrix(int,int);
};

```

Функция должна возвращать новую матрицу При разных размерах матриц должна порождаться искл. ситуация bad\_size