

Программа зачета по курсу
"Программирование на языке C++"
ФПМИ МФТИ

Даниил Гагаринов
github.com/yaishenka

Осень 2021

Содержание

Правила зачета	6
Первая итерация (святой рандом)	6
Вторая итерация	6
Третья итерация	6
Введение в язык	7
1 вопрос	7
2 вопрос	7
3 вопрос	7
4 вопрос	7
5 вопрос	7
6 вопрос	8
7 вопрос	8
8 вопрос	8
9 вопрос	9
10 вопрос	9
11 вопрос	9
12 вопрос	9
Введение в ООП	10
1 вопрос	10
2 вопрос	10
3 вопрос	10
4 вопрос	11
5 вопрос	11
6 вопрос	12
7 вопрос	12
8 вопрос	13
9 вопрос	13

10 вопрос	13
11 вопрос	14
12 вопрос	14
13 вопрос	15
14 вопрос	15
15 вопрос	16
16 вопрос	16
16 вопрос	16
Перегрузка операторов	18
1 вопрос	18
2 вопрос	18
3 вопрос	18
4 вопрос	19
5 вопрос	19
6 вопрос	20
7 вопрос	20
8 вопрос	21
8 вопрос	21
Наследование	23
1 вопрос	23
2 вопрос	23
3 вопрос	24
4 вопрос	25
4 вопрос	26
5 вопрос	26
6 вопрос	27
7 вопрос	28
8 вопрос	28

Виртуальные функции и тд и тп	30
1 вопрос	30
2 вопрос	30
3 вопрос	31
4 вопрос	31
5 вопрос	32
5 вопрос	32
6 вопрос	33
Введение в шаблоны	34
1 вопрос	34
2 вопрос	34
3 вопрос	35
4 вопрос	35
5 вопрос	35
6 вопрос	36
7 вопрос	37
8 вопрос	37
9 вопрос	38
10 вопрос	39
11 вопрос	39
Type-traits, variadic templates, правила вывода шаблонов	41
1 вопрос	41
2 вопрос	41
3 вопрос	41
4 вопрос	41
5 вопрос	41
6 вопрос	41
7 вопрос	42
8 вопрос	42

9 вопрос	42
10 вопрос	43
Исключения	44
1 вопрос	44
2 вопрос	44
3 вопрос	44
4 вопрос	45
5 вопрос	45
6 вопрос	45
7 вопрос	46
8 вопрос	46
9 вопрос	47
10 вопрос	47
11 вопрос	48
12 вопрос	48

Правила зачета

За зачет можно получить оценку от 1 до 10. Она будет весить 50% оценки курса.

Зачет будет проходить в несколько итераций.

Первая итерация (святой рандом)

Студенту выпадает случайный вопрос из каждой секции (всего 8 вопросов). У студента есть по 2 минуты на ответ на каждый вопрос (всего не более 20 минут (на всякий случай)). За каждый верный ответ студент получает +1 балл. За каждый неверный ответ или за неданный ответ на вопрос студент получает -1 (при этом меньше 0 упасть нельзя). Итого можно набрать от 0 до 8 баллов

Вторая итерация

Если студент набрал меньше 4 баллов, то он уходит с 0 за зачет. Если студент набрал 8 баллов, то он уходит с 10 за зачет. Далее студенту задается 8 - n вопросов (где n - число набранных баллов в первой итерации). Вопросы выбираются на усмотрение экзаменатора. Студент так же получает +1 за каждый верный ответ и -1 за неверный или не данный ответ, однако его изначальные n баллов не сгорают. Если экзаменатор понимает, что студент не ботал какую-то секцию (или не одну), то он может поставить 0 за зачет.

Третья итерация

Если вы согласны со своей оценкой, то вы идете домой. А если нет, то можете апеллировать к лектору. Апелляция может как повысить, так и понизить ваши баллы. Удачи!

Введение в язык

1 вопрос

Задача. Дан следующий код. Компилируется ли этот код? Если нет, то объясните почему

```
#include <iostream>

int& f(int x) {
    return x * x;
}

int main() {
    std::cout << f(5);
}
```

2 вопрос

Задача. Чему будет равна переменная c после выполнения следующего кода

```
int a = 1;
int b = 2;
int c = 0;
if (a > 1)
    if (b > 1) c = 5;
else
    c = 6;
```

3 вопрос

Задача. Что такое static_cast? Приведите пример использования

4 вопрос

Задача. Что такое reinterpret_cast? Приведите пример использования.

5 вопрос

Задача. Что такое c-style cast?

6 вопрос

Задача. Что выведется на экран, после выполнения следующего кода? Объясните почему так.

```
#include <iostream>

void f(int value) { value += 1; }

void g(int& value) { value += 1; }

int main() {
    int a = 10;
    int b = 10;

    int& x = a;
    int& y = b;

    f(a); f(b); g(a); g(b);
    f(x); f(y); g(x); g(y);

    std::cout << a << " " << b << " " << x << " " << y << std::endl;
}
```

7 вопрос

Задача. Есть ли в данном объявлении функции проблемы?

```
double Area(double Pi=3.14, int Radius);
```

8 вопрос

Задача. Корректен ли данный фрагмент кода? Если да, то чему равны значения всех переменных? Если же нет, то напишите, какие строки являются некорректными и почему.

```
int a = 3;
const int& b = a;
const int& c = 5;
int& d;
const int& e;
const int f = 6;
int& g = f;
```


9 вопрос

Задача. Корректен ли данный фрагмент кода? Если да, то чему равны значения s и f соответственно? Если же нет, то опишите, почему.

```
int& a = 3;
int b = 3;
int& c = b;
int* d = &c;
int*& e = c;
int*& f = d;
```

10 вопрос

Задача. Возникнет ли в данном коде СЕ? Какую из строк необходимо (если СЕ есть) закомментировать, чтобы избавиться от СЕ. Напомним, `auto` в данном контексте нужен, чтобы компилятор сам определил тип.

```
int*& arr_ref = new int& [10]; // (1)
auto arr_ref = new int& [10]; // (2)
```

11 вопрос

Задача. Что это за покемон? Он работает как нужно?

```
bool IsZero(int i) {
    return i&&&i;
}
```

12 вопрос

Задача. Чем отличается постфиксный и префиксный инкременты?

Введение в ООП

1 вопрос

Задача. Скомпилируется ли данный код?

```
class S {  
    int GetX() {  
        return x;  
    }  
private:  
    int x;  
};  
  
int main() {  
    S s;  
    std::cout << s.GetX();  
}
```

2 вопрос

Задача. Что такое ключевое слово mutable? Приведите пример, где его корректно использовать.

3 вопрос

Задача. Скомпилируется ли данный код?

```
struct S {  
    static int GetX() {  
        return x;  
    }  
private:  
    int x;  
};  
  
int main() {  
    S s;  
    std::cout << s.GetX();  
}
```

4 вопрос

Задача. Скомпилируется ли данный код?

```
struct S {  
    static int GetX() const {  
        return x;  
    }  
private:  
    static int x;  
};
```

```
int main() {  
    S s;  
    std::cout << s.GetX();  
}
```

5 вопрос

Задача. Скомпилируется ли данный код?

```
class Tree {  
public:  
    Node& GetRoot() {  
        return root_;  
    }  
private:  
    struct Node {  
        int val = 0;  
    };  
    Node root_;  
};
```

```
int main() {  
    Tree t;  
    t.GetRoot().val = 4;  
    std::cout << t.GetRoot().val << '\n';  
}
```

6 вопрос

Задача. Скомпилируется ли данный код? Объясните с точки зрения ООП, в чем проблема такого кода?

```
class Tree {
private:
    struct Node;
public:
    Node& GetRoot() {
        return root_;
    }
private:
    struct Node {
        int val = 0;
    };
    Node root_;
};

int main() {
    Tree t;
    t.GetRoot().val = 4;
    std::cout << t.GetRoot().val << '\n';
}
```

7 вопрос

Задача. Есть ли проблемы в следующем коде, и если да, то какие?

```
class C {
public:
    C(int size) {
        arr_ = new int[size];
    }
private:
    int* arr_ = nullptr;
};

int main() {
    C c(10);
}
```

8 вопрос

Задача. Есть ли проблемы в следующем коде, и если да, то какие?

```
class C {  
    public:  
        C(int size) {  
            arr_ = new int[size];  
        }  
    private:  
        ~C() {  
            delete[] arr_;  
        }  
  
        int* arr_ = nullptr;  
};  
  
int main() {  
    C c(10);  
}
```

9 вопрос

Задача. Есть ли проблемы в следующем коде, и если да, то какие?

```
class C {  
    public:  
        ~C(int a) {  
            std::cout << a;  
        }  
};  
  
int main() {  
    C c;  
}
```

10 вопрос

Задача. Есть ли проблемы в следующем коде, и если да, то какие?

```
class C {  
    public:  
        ~C(int a) {  
            std::cout << a;  
        }  
};  
  
int main() {  
    C c;  
}
```

11 вопрос

Задача. Есть ли проблемы в следующем классе, и если да, то какие?

```
class String {  
    public:  
        String(int size) {  
            str_ = new char[size];  
        }  
  
        ~String() {  
            delete[] str_;  
        }  
    private:  
        char* str_ = nullptr;  
};
```

12 вопрос

Задача. Есть ли проблемы в следующем классе, и если да, то какие и как их исправить?

```
class String {  
    public:  
        /*  
            Предположим что основные конструкторы и деструктор написаны  
        */  
  
        String& operator=(const String& other) {
```

```
        str_ = other.str_;
    }
private:
    char* str_ = nullptr;
};
```

13 вопрос

Задача. Есть ли проблемы в следующем классе, и если да, то какие и как их исправить?

```
struct S {
    void GetX() const {
        number_of_requests_to_x += 1;
        return x;
    }

    int x;
    int number_of_requests_to_x = 0;
};
```

14 вопрос

Задача. Как переписать следующий код, чтобы он был компактней?

```
class C {
public:
    C(int x, int y, int z, int w, int t) {
        x_ = x;
        y_ = y;
        z_ = z;
        w_ = w;
        t_ = t;
    }

private:
    int x_;
    int y_;
    int z_;
    int w_;
    int t_;
};
```

15 вопрос

Задача. Как определить метод [] для следующего класса?

```
class Array {
public:
    explicit Array(size_t size) {
        arr_ = new int[size];
    }

    ~Array() { delete[] arr_; }

private:
    int* arr_;
};
```

16 вопрос

Задача. Как упростить следующий код?

```
class Student {
public:
    Student(const Student& other) {
        full_name_ = other.full_name_;
        group_name_ = other.group_name_;
        grade_ = other.grade_;
    }

private:
    std::string full_name_;
    std::string group_name_;
    size_t grade_;
};
```

16 вопрос

Задача. Как избежать копипасты в следующем коде?


```
class Student {
public:
    explicit Student(std::string full_name): full_name_(full_name) {
        if (full_name_ == "Гений") {
            grade_ = 10;
        }
    }

    Student(std::string full_name, std::string group_name): full_name_(full_name), group_name_(group_name) {
        if (full_name_ == "Гений") {
            grade_ = 10;
        }
    }

private:
    std::string full_name_;
    std::string group_name_;
    size_t grade_;
};
```

Перегрузка операторов

1 вопрос

Задача. Есть класс `MyInt`, который устроен следующим образом:

```
class MyInt {
    MyInt(int value): value_(value) {}
private:
    int value_;
};
```

Реализуйте операторы `*` и `*=` (на другой объект типа `MyInt`). Уточните для каждого из них где вы объявили этот оператор (в классе или нет). Объясните свой выбор.

2 вопрос

Есть структура `Array`, которая устроена следующим образом:

```
struct Array {
    int* arr_;
}
```

Реализуйте оператор присваивания копированием для этой структуры.

3 вопрос

Задача. Есть класс `MyInt`, который устроен следующим образом:

```
class MyInt {
    MyInt(int value): value_(value) {}
private:
    int value_;
};
```

Реализуйте для него операторы сравнения. Укажите где вы объявляете эти операторы (внутри класса или нет)

4 вопрос

Задача. Есть класс `MyInt`, который устроен следующим образом:

```
class MyInt {
    MyInt(int value): value_(value) {}
private:
    int value_;
};
```

Реализуйте для него операторы префиксного и постфиксного инкрементов

5 вопрос

Задача. Есть класс `String`, который устроен следующим образом:

```
class String {
public:
    friend String operator+(const String& first, const String& second);

    String(const char* str) {
        str_.resize(strlen(str));
        std::copy(str, str + strlen(str), str_.begin());
    }

    String& operator+=(const String& other) {
        *this = other + *this;
        return *this;
    }
private:
    std::vector<char> str_;
};
```

```
String operator+(const String& first, const String& second) {
    String result = first;
    for (char c : second.str_) {
        first.str_.push_back(c);
    }
}
```

```
    return result;
}
```

Какие проблемы есть с реализацией операторов `+` и `+=`? Как их исправить?

6 вопрос

Задача. Есть класс `IntPtrter`

```
class IntPtrter {
    MyInt(int* ptr): ptr_(ptr) {}
private:
    int* ptr_;
};
```

Реализуйте для него операторы `*` и `->`, которые бы работали аналогично просто `int*`

7 вопрос

Задача. Какие есть проблемы в следующем коде?

```
class MyInt {
    MyInt(int value): value_(value) {}

    explicit bool() {
        return value_ != 0;
    }

private:
    int value_;
};

int main() {
    MyInt a(10);
    if (a) {
        std::cout << "true";
    } else {
        std::cout << "false";
    }
}
```

```
    }  
}
```

8 вопрос

Задача. Что выведется на экран после выполнения кода ниже? Объясните почему так. Как исправить этот код так, чтобы стало меньше копирований?

```
#include <iostream>  
  
class C {  
public:  
    C() = default;  
  
    C(const C& other) {std::cout << 1 << std::endl; }  
  
    C& operator=(const C& other) {  
        std::cout << 2 << std::endl;  
        return *this;  
    }  
  
    C& operator+=(const C& other) { return *this; }  
};  
  
C operator+(const C& first, const C& second) {  
    C copy = first;  
    return copy += second;  
}  
  
int main() {  
    C a, b;  
    C c = a + b;  
}
```

8 вопрос

Задача. Перед вами класс `Int` и код, взаимодействующий с ним. Что может пойти не так? Исправится ли ошибка, если убрать слово `explicit`? Как исправить код класса `Int`?

```
#include <iostream>

class Int {
public:
    Int() = default;
    explicit Int(int x) : x_(x) {}

    bool operator<(const Int& other) const {
        return x_ < other.x_;
    }

private:
    int x_ = 0;
};

int main() {
    Int a;
    if (3 < a) { std::cout << "less\n"; }
}
```

Наследование

1 вопрос

Задача. Что выведется на экран при исполнении следующего кода? Подробно объясните.

```
#include <iostream>

class A {
public:
    A() {std::cout << "create A" << std::endl; }
    ~A() {std::cout << "destroy A" << std::endl; }
};

class C {
public:
    C() {std::cout << "create C" << std::endl; }
    ~C() {std::cout << "destroy C" << std::endl; }
};

class B: public A {
public:
    B() {std::cout << "create B" << std::endl; }
    ~B() {std::cout << "destroy B" << std::endl; }

private:
    C c;
};

int main() {
    B b;
    A a = b;
}
```

2 вопрос

Задача. Есть ли проблемы в следующем коде? Если да, то расскажите как их исправить

```
#include <iostream>
```

```
class A {
public:
    explicit A(int x): x_(x) {}
    int GetX() const { return x_; }
private:
    int x_;
};

class B: A {
public:
    explicit B(int x): A(x) {}
};

int main() {
    B b(10);
    std::cout << b.GetX();
}
```

3 вопрос

Задача. Есть следующий код:

```
#include <iostream>

class Granny {
protected:
    int g;
    friend int main();
};

class Mom: private Granny {
private:
    int m;
    friend class Son;
};

class Son: public Mom {
public:
    void f() {
```



```
    Granny granny; // 1
    std::cout << g; // 2
}
};

int main() {
    Son s;
    std::cout << s.m << s.g; // 3
}
```

В каких строках (1, 2 и 3) будет ошибка ркомпиляции? Почему? Объясните подробно.

4 вопрос

Задача. Есть следующий код:

```
#include <iostream>

class Granny {
protected:
    int g;
    friend int main();
};

class Mom: private Granny {
private:
    int m;
};

class Son: public Mom {
public:
    void f() {
        Granny granny; // 1
        std::cout << g; // 2
    }
};

int main() {
    Son s;
```

```
std::cout << s.m << s.g; // 3
}
```

В каких строках (1, 2 и 3) будет ошибка компиляции? Почему? Объясните подробно.

4 вопрос

Задача. Есть следующий код:

```
#include <iostream>

class Granny {
protected:
    int g;
    friend int main();
};

class Mom: private Granny {
private:
    int m;
};

class Son: public Mom {
public:
    void f() {
        ::Granny granny; // 1
        std::cout << g; // 2
    }
};

int main() {
    Son s;
    std::cout << s.m << s.g; // 3
}
```

В каких строках (1, 2 и 3) будет ошибка компиляции? Почему? Объясните подробно.

5 вопрос

Задача. Что выведет следующий код и почему?

```
#include <iostream>

struct B {
    B() {
        std::cout << "B constructed\n";
    }
    ~B() {
        std::cout << "B destroyed\n";
    }
};

struct D : B {
    D() {
        std::cout << "D constructed\n";
    }
    ~D() {
        std::cout << "D destroyed\n";
    }
};

int main() {
    B* b = new D();
    delete b;
}
```

6 вопрос

Задача. В каких из строк (1, 2 и 3) возникнут ошибки компиляции и почему?

```
#include <iostream>

struct Granny {
    int g = 0;
};

class Mom: public Granny {
    int m = 1;
};

class Dad: private Granny {
```

```
    int d = 2;
};

class Son: private Mom, public Dad {
    int s = 3;
};

int main() {
    Son s;

    std::cout << s.g; // 1
    std::cout << static_cast<Mom&>(s).m << static_cast<Mom&>(s).g; // 2
    std::cout << static_cast<Dad&>(s).d << static_cast<Dad&>(s).g; // 3
}
```

7 вопрос

Задача. Что такое проблема ромбовидного наследования? Приведите пример. Как решается проблема ромбовидного наследования в плюсах?

8 вопрос

Задача. Что выведется на экран после выполнения следующего кода? Почему?

```
#include <iostream>

struct Granny {
    int a = 10;
};

struct Mom: virtual Granny {
    Mom() {a = 100; }
};

struct Dad: virtual Granny {
    Dad() {a = 1000; }
};

struct Son: Dad, Mom {
```

```
};
```

```
int main() {  
    Son s;  
    std::cout << s.a;  
}
```

Виртуальные функции и тд и тп

1 вопрос

Задача. Скомпилируется ли данный код? Если нет, то объясните почему, если да, то что выведется на экран и почему?

```
#include <iostream>

struct Base {
    virtual void foo() const {
        std::cout << 1;
    }
};

struct Derived: Base {
    virtual void foo() {
        std::cout << 2;
    }
};

int main() {
    Derived d;
    d.foo();
    Base& b = d;
    b.foo();
}
```

2 вопрос

Задача. Скомпилируется ли данный код? Если нет, то объясните почему, если да, то что выведется на экран и почему? Что делает override?

```
#include <iostream>

struct Base {
    virtual void foo() const {
        std::cout << 1;
    }
}
```

```
};

struct Derived: Base {
    virtual void foo() override {
        std::cout << 2;
    }
};

int main() {
    Derived d;
    d.foo();
    Base& b = d;
    b.foo();
}
```

3 вопрос

Задача. Зачем нужны виртуальные деструкторы? Приведите пример. Существуют ли виртуальные конструкторы?

4 вопрос

Задача. Скомпилируется ли данный код? Если нет, то объясните почему, если да, то что выведется на экран и почему? Что делает override?

```
#include <iostream>

struct Base {
    virtual void foo() const {
        std::cout << 1;
    }
};

struct Derived: Base {
    virtual void foo() override {
        std::cout << 2;
    }
};
```

```
int main() {  
    Derived d;  
    d.foo();  
    Base& b = d;  
    b.foo();  
}
```

5 вопрос

Задача. Скомпилируется ли данный код? Если нет, то объясните почему, если да, то что выведется на экран и почему? Что делает final?

```
#include <iostream>  
  
struct Granny {  
    virtual void foo() {std::cout << "granny"; }  
};  
  
struct Mom: Granny {  
    void foo() final {std::cout << "mom"; }  
};  
  
struct Son: Mom {  
    void foo() override {std::cout << "son"; }  
};  
  
int main() {  
    Son s;  
    s.foo();  
}
```

5 вопрос

Задача. Скомпилируется ли данный код? Если нет, то объясните почему.

```
#include <iostream>  
  
struct Base {  
    virtual int foo() = 0;  
};
```



```
};

int Base::foo() {
    return 5
}

struct Derived: Base {};

int main() {
    Derived d;
    std::cout << d.foo();
}
```

6 вопрос

Задача. Скомпилируется ли данный код? Если нет, то объясните почему, если да, то что выведется на экран?

```
#include <iostream>

struct Base {
    int foo() {return 1; }
};

struct Derived: Base {
    int foo() {return 2; }
};

int main() {
    Derived d;
    Base& b = d;
    Derived& new_d = dynamic_cast<Derived&>(b);
    std::cout << new_d.foo();
}
```

Введение в шаблоны

1 вопрос

Задача. Оказалось, что следующий код вызывает ошибку компиляции. Объясните почему. Как исправить код?

```
#include <iostream>

template <size_t N>
struct S {
    size_t foo() {
        return N;
    }
};

int main() {
    size_t n = 5;
    S<n> s;
    std::cout << s.foo();
}
```

2 вопрос

Задача. Компилируется ли следующий код? Если нет, то объясните почему.

```
#include <iostream>

template <size_t N>
struct S {
    size_t foo() {
        return N;
    }
};

int main() {
    size_t t = 5;
    const size_t n = t;
    S<n> s;
```

```
std::cout << s.foo();  
}
```

3 вопрос

Задача. Есть ли проблемы в следующем коде? Если да, то объясните какие

```
template <typename T, typename U>  
const T& GetMax(const T& a, const U& b) {  
    return a > b ? a : b;  
}  
  
int main() {  
    GetMax(1, 2.0);  
}
```

4 вопрос

Задача. Работает ли следующий код? Если нет, то объясните почему

```
template <typename T>  
struct Vector {  
    template <typename U>  
    void push_back(const U&);  
  
    T value;  
}  
  
template <typename T, typename U>  
template <typename U>  
void Vector<T>::push_back(const U&) {  
    ...  
}
```

5 вопрос

Задача. Работает ли следующий код? Если нет, то объясните почему. Если да, то что выведется на экран?

```
#include <iostream>

template <typename T>
void f(T x) {
    std::cout << 1;
}

template <typename T>
void f(T& x) {
    std::cout << 2;
}

int main() {
    int x = 0;
    int& y = x;
    f(y);
}
```

6 вопрос

Задача. Что выведется на экран?

```
#include <iostream>

template <typename T, U>
void f(T x, U y) {
    std::cout << 1;
}

template <typename T>
void f(T x, T y) {
    std::cout << 2;
}

void f(int x, double y) {
    std::cout << 3;
}

int main() {
```

```
    f(0, 0);  
}
```

7 вопрос

Задача. Есть следующий код

```
#include <iostream>  
  
template <typename T, typename U>  
void f(T, U) {  
    std::cout << 1;  
}  
  
template <typename T>  
void f(T, T) {  
    std::cout << 2;  
}  
  
template <>  
void f(int, int) {  
    std::cout << 3;  
}  
  
int main() {  
    f(0, 0);  
}
```

Что выведется на экран?

8 вопрос

Задача. Есть следующий код

```
#include <iostream>  
  
template <typename T, typename U>  
void f(T, U) {  
    std::cout << 1;
```

```
}

template <>
void f(int, int) {
    std::cout << 3;
}

template <typename T>
void f(T, T) {
    std::cout << 2;
}

int main() {
    f(0, 0);
}
```

Что выведется на экран?

9 вопрос

Задача. Есть следующий код

```
#include <iostream>

template <typename T, typename U>
void f(T, U) {
    std::cout << 1;
}

void f(int, int) {
    std::cout << 3;
}

template <typename T>
void f(T, T) {
    std::cout << 2;
}

int main() {
```

```
f(0, 0);  
}
```

Что выведется на экран?

10 вопрос

Задача. Оказалось, что следующий код приводит к СЕ. Объясните почему. Что надо исправить, чтобы СЕ не было?

```
template <typename T>  
struct S {  
    using X = T;  
};
```

```
template <>  
struct S<int> {  
    static int X;  
};
```

```
int a = 0;
```

```
template <typename T>  
void f() {  
    S<T>::X * a;  
}
```

```
int main() {  
    f<double>();  
}
```

11 вопрос

Задача. Оказалось, что следующий код приводит к СЕ. Объясните почему. Что надо исправить, чтобы СЕ не было?

```
template <typename T>  
struct S {  
    template <int M, int N>
```

```
    struct A{};
}

template <>
struct S<int> {
    static const int A = 0;
}

template <typename T>
void f() {
    S<T>::A<1,2> a;
}
```


Type-traits, variadic templates, правила вывода шаблонов

1 вопрос

Задача. Реализуйте `std::is_same`. Объясните как это работает

2 вопрос

Задача. Реализуйте `std::remove_reference`. Объясните как это работает

3 вопрос

Задача. Реализуйте `std::remove_const`. Объясните как это работает

4 вопрос

Задача. Допустим у нас есть реализация `std::is_same<U,V>`. Реализуйте удобный псевдоним `std::is_same_v`.

5 вопрос

Задача. Реализуйте функцию `print`, которая умеет принимать любое количество любых аргументов и выводит их в поток (считаем что оператор «`u`» у всех аргументов точно есть)

6 вопрос

Задача. Напишите type_trait `is_same_many` (примеры вызовов смотрите ниже, вместо многоточия должен быть ваш код). Разрешается пользоваться `is_same` (и `is_same_v`) соответственно. Вспомните про вариадика!

```
template <...>
struct is_same_many {
    const static bool value = ...;
}
```

```
template <...>
struct is_same_many {
    const static bool value = ...;
```

```

}

int main() {
    is_same_many<int, int, int>::value; // true
    is_same_many<double, int, int, std::string>::value; // false
    is_same_many<double, double, double, double, long long int>::value; // false
}

```

7 вопрос

Задача. Есть такая функция `f` и `main`. Какой тип выведется при каждом вызове?

```

template <typename T>
void f(T x) {}

int main() {
    int x = 0; int& y = x; const int& z = y;
    f(x); f(y); f(z);
}

```

	T
x	
y	
z	

8 вопрос

Задача. Как понять какой тип вывел компилятор при шаблонной подстановке? Приведите пример кода.

9 вопрос

Задача. Есть такая функция `f` и `main`. Какой тип выведется при каждом вызове?

```

template <typename T>
void f(T& x) {}

int main() {
    int x = 0; int& y = x; const int& z = y; const int c = 0;
    f(x); f(y); f(z); f(c);
}

```

	T
x	
y	
z	
c	

10 вопрос

Задача. Есть такая функция `f` и `main`. Какой тип выведется при каждом вызове?

```
template <typename T>
void f(const T& x) {}
int main() {
    int x = 0; int& y = x; const int& z = y; const int c = 0;
    f(x); f(y); f(z); f(c);
}
```

	T
x	
y	
z	
c	

Исключения

1 вопрос

Задача. Отработает ли следующий код без ошибок? Если нет, то объясните почему.

```
int main() {  
    try {  
        int a = 10 / 0;  
    } catch(...) {  
    }  
}
```

2 вопрос

Задача. Отработает ли следующий код без ошибок? Если нет, то объясните почему.

```
int main() {  
    int* a = new int[10];  
    try {  
        int b = a[10];  
    } catch(...) {  
    }  
}
```

3 вопрос

Задача. Отработает ли следующий код без ошибок? Если нет, то объясните почему.

```
int main() {  
    std::vector<int> v(10, 0);  
    try {  
        int b = v[10];  
    } catch(...) {  
    }  
}
```

4 вопрос

Задача. Отработает ли следующий код без ошибок? Если нет, то объясните почему.

```
int main() {  
    std::vector<int> v(10, 0);  
    try {  
        int b = v.at(10);  
    } catch(...) {  
    }  
}
```

5 вопрос

Задача. Является ли этот код безопасным относительно исключений?

```
int main() {  
    int* a = new int[10];  
}
```

6 вопрос

Задача. Является ли этот код безопасным относительно исключений?

```
int main() {  
    try {  
        throw std::out_of_range("aaaa");  
    } catch (std::exception& ex){  
        throw 1;  
        std::cout << 1;  
    } catch (std::out_of_range& ex) {  
        std::cout << 2;  
    } catch (...) {  
        std::cout << 3;  
    }  
}
```

7 вопрос

Задача. Что выведется на экран после исполнения этого кода?

```
#include <iostream>

struct S {
    S() {std::cout << 1; }
    S(const S&) {std::cout << 2; }
    ~S() {std::cout << 3; }
};

void f() { S s; throw s; }

int main() {
    try {
        f();
    } catch (S s) {
    }
}
```

8 вопрос

Задача. Что выведет следующий код?

```
struct Base {};
struct Derived: Base {};
void f() {
    try {
        Derived d;
        throw d;
    } catch (Base& b) {
        throw b;
    }
}

int main() {
    try {
        f();
    } catch (Derived& d) {
        std::cout << "derived";
    } catch (Base& b) {
```

```
    std::cout << "base";  
}  
}
```

А что выведется, если переписать функцию f:

```
void f() {  
    try {  
        Derived d;  
        throw d;  
    } catch (Base& b) {  
        throw; // тут поменяли!  
    }  
}
```

9 вопрос

Задача. В чем проблема следующего кода?

```
struct S {  
    int* p = nullptr;  
    S(): p(new int(5)) {  
        throw 1;  
    }  
  
    ~S() {  
        delete p;  
    }  
}  
  
int main() {  
    try {  
        S s;  
    } catch(...) {  
  
    }  
}
```

10 вопрос

Задача. Что будет, если исполнить следующий код?

11 вопрос

Задача. Что будет выведено на экран?

```
#include <iostream>

int main() {
    std::cout << noexcept(1 / 0);
}
```

12 вопрос

Задача. Что произойдет при выполнении следующего кода?

```
struct S {
    int x = 0;
    S(int x): x(x) {}
    ~S() {
        if (x == 0) {
            throw 1;
        }
    }
}

void g() { S s(0); }

void f() { S s(0); g(); }

int main() {
    try {
        f();
    } catch (...) {
        std::cout << "caught\n";
    }
}
```