**Вопрос 1**

Что будет напечатано, если letter примет значение 'a'?

switch(letter)  
{  
case 'a': printf("Отпусти свой ум. n"); break;  
case 'e': printf("Наблюдайте за вашим телом, если хотите, чтобы ваш ум  
работал правильно.n"); break;  
case 'c': printf("Мудрость не скажет того, что противно природе.n"); break;  
case 'n': printf(" Удача - это постоянная готовность использовать  
шанс.n");break;  
default: printf("Юмор - это спасательный круг на волнах жизни.n");  
}

Правильный ответ: Отпусти свой ум.

**Вопрос 2**

Qt — фреймворк для разработки кроссплатформенного программного обеспечения на языке программирования C++.

Фре́ймворк — программная платформа, определяющая структуру программной системы; программное обеспечение, облегчающее разработку и объединение разных компонентов большого программного проекта.

**Вопрос 3**

UML (англ. Unified Modeling Language — унифицированный язык моделирования) — язык графического описания для объектного моделирования в области разработки программного обеспечения, для моделирования бизнес-процессов, системного проектирования и отображения организационных структур.

**Вопрос 4**

Абстрактный класс в объектно-ориентированном программировании — базовый класс, который не предполагает создания экземпляров. Абстрактные классы реализуют на практике один из принципов ООП — полиморфизм. Абстрактный класс может содержать (и не содержать) абстрактные методы и свойства.

**Вопрос 5**

Актер в диаграмме прецедентов – это любая сущность, которая выполняет роль в одной данной системе. Это может быть человек, организация или внешняя система и обычно рисуется как скелет.

**Вопрос 6**

Шаблон проектирования или паттерн в разработке программного обеспечения — повторяемая архитектурная конструкция, представляющая собой решение проблемы проектирования в рамках некоторого часто возникающего контекста.

**Вопрос 7**

В каких выражениях правильно определен метод класса Ping?

class Ping

{ public: float f, d;

int a;

void MathFunc(double D);

};

Варианты ответов:

1. Ping::MathFunc(double D) {f = D;}

2. void Ping::MathFunc(double D) {f = D;}

3. void Ping::MathFunc(double D) {this->f = D;}

Правильный ответ: 2, 3

**Вопрос 8**

В каких строках будут ошибки при компиляции данного кода?

#include <iostream>

using namespace std;

class Test

{

public:

bool flag;

Test () { flag = false; }

static int ObjectCount; // 1

static void POut () // 2

{

cout << flag; // 3

cout << ObjectCount; // 4

}

};

int Test::ObjectCount = 0; // 5

int main ()

{

Test T;}

Правильный ответ: 3

**Вопрос 9**

В каких строках допущены ошибки:

int main()

{

int intvar;

float flovar;

int\* intptr;

float\* floptr;

void \* voidptr;

intptr = &intvar; //1

intptr = &flovar; //2

floptr = &intvar; //3

floptr = &flovar; //4

voidptr =&intvar; //5

voidptr =&flovar; //6

}

Правильный ответ: 2, 3

**Вопрос 10**

В какой из перечисленных строк произойдет ошибка компиляции:

class Base {

public:

void method1(); //3

protected:

void method2();

private:

void method3();

};

class Child : public Base {

protected:

void method1() { }

void method2() { }

void method3() { } //4

};

int main() {

Base\* base = new Child();

base->method1(); // 1

base->method2(); // 2

return 0;

}

Правильный ответ: 2.

**Вопрос 11**

В какой строке данного фрагмента кода допущена ошибка:

int a = 3;

int i; // 2

for (i = 0; i < 3; i++) { // 3

a++; // 4

}

Правильный ответ: ошибок нет

**Вопрос 12**

В какой строке нижеприведенного кода создается пустое множество?

1: #include <iostream>

2: #include <set>

3: using namespace std;

4: typedef set<int. less<int> > set\_i ;

5: set\_i::iterator i;

6: int main(){

7: int a[4] = {4, 2, 1, 2};

8: set\_i s1;

9: set\_i s2(a, a + 4);

10: set\_i s3(s2);

11: s2.insert(10);

12: s2.insert(6);

13: return 0;

14: }

Правильный ответ: 8

**Вопрос 13**

В каком из вариантов объявлен двумерный массив в С ++?

A. int anarray [20] [20];

B. int array [20, 20];

C. array anarray [20] [20];

D. char array [20];

Правильный ответ: А

**Вопрос 14**

В каких из перечисленных фрагментов кода указатели объявлены правильно?

A. int \*p4

B. float p5\*;

C. \*char p2;

D. int z = 20; int\* p3 = &z

E. int \* p1;

Правильный ответ: А, D, E

**Вопрос 15**

В каком случае описание класса верно?

1. class A

{

int x;

public:

int F (int x){ if(!x)return (\*this).x;}

};

2. class A

{

int x;

public:

int getNew(int x){

A a;

this = &a;

return this;

}

};

Правильный ответ: 1

**Вопрос 16**

120. В каком файле заголовков определен объект cout?

A. stream.h

B. sysutils.hpp

C. cout.h

D. iostream.h

Правильный ответ: D

**Вопрос 17**

Известно, что в классе A определен один публичный конструктор A(int);. Выберите из предложенных выражений компилируемые:

1. A a = new A(4);

2. A a(4);

Правильный ответ: 2

**Вопрос 18 - 19**

Переменная unsigned с n-ным количеством бит имеет диапазон от 0 до (2n)-1.

**Вопрос 20**

Если есть два объявления int qwerty; int QWERTY; какое из утверждений верно?

Правильный ответ: Оба объявления правильные

**Вопрос 21**

В программе на языке С ++ объявлены такие переменные int x, y; Выражение позволяет вычислить остаток от деления этих переменных?

Если будут ответы да нет, то Правильный ответ: нет.

Если будут перечислены операции, то Правильный ответ: %

**Вопрос 22**

Функциональные особенности Стандартной Библиотеки объявляются внутри пространства имен std. Стандартная библиотека шаблонов (STL) — подмножество стандартной библиотеки C++ и содержит контейнеры, алгоритмы, итераторы, объекты-функции и т. д.

**Вопрос 23**

Разница между фактическими параметрами и формальными параметрами заключается в том, что фактические параметры - это значения, которые передаются функции при ее вызове, в то время как формальные параметры - это переменные, определенные функцией, которая получает значения при вызове функции.

**Вопрос 24**

Верно ли составлен данный код? Что является результатом работы функции f()?

class A {

public:

int inc(int x) { return ++x; };

int inc(short x) { return x + 2; };

};

void f()

{

A obj;

int y = 5;

cout << obj.inc(y);

}

Правильный ответ: 6

**Вопрос 25**

Диаграмма классов является ключевым элементом в объектно-ориентированном моделировании. На диаграмме классы представлены в рамках, содержащих три компонента:

В верхней части написано имя класса. Имя класса выравнивается по центру и пишется полужирным шрифтом. Имена классов начинаются с заглавной буквы. Если класс абстрактный — то его имя пишется полужирным курсивом.

**Вопрос 26**

Виртуальная функция — это функция, которая определяется в базовом классе, а любой порожденный класс может ее переопределить. Виртуальная функция вызывается только через указатель или ссылку на базовый класс. Указатель на базовый класс может указывать либо на объект базового класса, либо на объект порожденного класса.

**Вопрос 27,29,36**

Деструктор — это специальный тип метода класса, который выполняется при удалении объекта класса. В то время как конструкторы предназначены для инициализации класса, деструкторы предназначены для очистки памяти после него.

* деструктор должен иметь то же имя, что и класс, со знаком тильда (~) в самом начале;
* деструктор не может принимать аргументы;
* деструктор не имеет типа возврата.

**Вопрос 30, 44, 59**

Дружественные функции - это функции, которые не являются членами класса, однако имеют доступ к его закрытым членам - переменным и функциям, которые имеют спецификатор private. При этом для дружественных функций не важно, определяются они под спецификатором public или private. Для них это не имеет значения.

**Вопрос 31, 39, 40, 46**

Модификаторы доступа — это ключевые слова, которые задают объявленный уровень доступности члена или типа.

Public, private и protected — это модификаторы доступа.

Public - доступ открыт всем другим классам, кто видит определение данного класса.

Protected — доступ открыт классам, производным от данного. То есть, производные классы получают свободный доступ к таким свойствам или метода. Все другие классы такого доступа не имеют.

Private — доступ открыт самому классу (т.е. функциям-членам данного класса) и друзьям (friend) данного класса - как функциям, так и классам. Однако производные классы не получают доступа к этим данным совсем. И все другие классы такого доступа не имеют.

**Вопрос 32**

Выберите варианты правильного объявления классов?

1. class A { A\* pNext; };

2. class A { int Age; void Func(int x){ Age = x;} };

3. class A { void F(); public: int Func(int x){ F(); };

Правильный ответ: 1, 2

**Вопрос 33, 35, 37, 107, 139**

Конструктор — это особый тип метода класса, который автоматически вызывается при создании объекта этого же класса. Конструкторы обычно используются для инициализации переменных-членов класса значениями, которые предоставлены по умолчанию/пользователем, или для выполнения любых шагов настройки, необходимых для используемого класса (например, открыть определенный файл или базу данных).

Конструкторы имеют определенные правила их именования:

* конструкторы всегда должны иметь то же имя, что и класс (учитываются верхний и нижний регистры);
* конструкторы не имеют типа возврата (даже void).

**Вопрос 38**

Инкапсуляция снижает скорость доступа к данным. Запрет на прямой доступ к полям класса извне приводит к необходимости создания и использования методов доступа. И написание, и компиляция, и исполнение методов доступа сопряжены с дополнительными расходами.

**Вопрос 41**

Фу́нкция в программировании, или подпрограмма — фрагмент программного кода, к которому можно обратиться из другого места программы. В большинстве случаев с функцией связывается идентификатор[en], но многие языки допускают и безымянные функции. С именем функции неразрывно связан адрес первой инструкции (оператора), входящей в функцию, которой передаётся управление при обращении к функции. После выполнения функции управление возвращается обратно в адрес возврата — точку программы, где данная функция была вызвана.

**Вопрос 42**

Выберите правильное объявление константы pi:

1. const float pi = 3.14;

2. float pi = (const) 3.14;

3. const float pi 3.14;

Правильный ответ: 1

**Вопрос 43**

class FirstClass // базовый класс

**……..…..…**

class SecondClass : public FirstClass   // производный класс

**………………**

**Вопрос 45**

Выберите правильное утверждение.

Если класс, производный от абстрактного, не определяет все чисто виртуальные функции, он является:

Правильный ответ: производным

**Вопрос 47**

Выберите правильный вызов функции-члена класса С через указатель "pmfn"

class Class1

{

public:

void func(); };

void (Class1::\*pmfn)() = &Class1::func;

Class1 \*c;

Ответ:

1. c->pmfn();

2. (c.\*pmfn)();

3. c.\*pmfn();

4. c->\*pmfn();

5. (c->\*pmfn)();

Правильный ответ: 1

**Вопрос 48**

Вызовет ли данный код ошибку компиляции?

class Rectangle{ public: int a,b; int sum(); int square(); ~rect(); };

1. да, имя деструктора не может начинаться с маленькой буквы
2. нет, все записано верно
3. да, имя деструктора должно совпадать с именем класса

Правильный ответ: 3

**Вопрос 49**

Сase объявляется с использованием ключевого слова case и имеет константное выражение. Константное выражение — это то, которое генерирует константное значение, другими словами: либо литерал (например, 5), либо перечисление (например, COLOR\_RED), либо константу (например, переменную const x).

Типичными CASE-инструментами являются:

* инструменты управления конфигурацией;
* инструменты моделирования данных;
* инструменты анализа и проектирования;
* инструменты преобразования моделей;
* инструменты редактирования программного кода;
* инструменты рефакторинга кода;
* генераторы кода;
* инструменты для построения UML-диаграмм.

**Вопрос 50**

Оператор default выполняется, если не найдено соответствий, default необязателен и, если его нет, то в случае отсутствия совпадений ничего не происходит. Когда обнаруживается совпадение, операторы, ассоциированные с соответствующим case, выполняются до тех пор, пока не встретится оператор break. В случае default (или последнего case, если отсутствует default), оператор switch заканчивает работу при обнаружении конца

**Вопрос 51**

Диаграмма классов — структурная диаграмма языка моделирования UML, демонстрирующая общую структуру иерархии классов системы, их коопераций, атрибутов (полей), методов, интерфейсов и взаимосвязей между ними. Широко применяется не только для документирования и визуализации, но также для конструирования посредством прямого или обратного проектирования.

**Вопрос 52**

Диаграмма вариантов использования в UML — диаграмма, отражающая отношения между актерами и прецедентами и являющаяся составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне.

Прецедент — возможность моделируемой системы (часть её функциональности), благодаря которой пользователь может получить конкретный, измеримый и нужный ему результат. Прецедент соответствует отдельному сервису системы, определяет один из вариантов её использования и описывает типичный способ взаимодействия пользователя с системой. Варианты использования обычно применяются для спецификации внешних требований к системе.

**Вопрос 53**

Для доступа к членам объекта через сам объект используется оператор «точка» (.). Если же используется указатель на объект, тогда необходимо использовать оператор «стрелка» (—>). Использование операторов «точка» и «стрелка» аналогично их использованию для структур и объединений.

**Вопрос 54**

Для чего используется ключевое слово finally в С++?

Правильный ответ: такого ключевого слова нет в С++

**Вопрос 55**

Встроенные функции – это функции, которые уже содержатся в табличном процессоре и выполняют различные вычисления автоматически при их вызове из библиотеки функций. Каждая функция имеет свое собственное имя, которое необходимо для ее вызова.

**Вопрос 56**

Оператор continue используется только в циклах. В операторах for, while, do while, оператор continue выполняет пропуск оставшейся части кода тела цикла и переходит к следующей итерации цикла.

**Вопрос 57**

Допускается ли перегрузка деструкторов?

Правильный ответ: Нет.

**Вопрос 58**

Допустимо ли определение чисто виртуального метода следующим образом:

class Abstract {

public:

virtual void pureVirtual() = 0 {

// реализация

}};

Правильный ответ: да

**Вопрос 60**

Если i=3, какой будет результат ?

if (i == 4) cout << "aaa";

else if (i == 3) cout << "bbb";

else if (i != 3) cout << "ccc";

Ответ:

1. aaa

2. bbb

3. ccc

4. aaaccc

5. bbbccc

6. ошибка компиляции

Правильный ответ: 2

**Вопрос 61**

Если записано

class A { public: void f() { cout << 1; } };

class B : public A

{ public: void f() { cout << 2; } };

то что будет напечатано?

B b; A& a=b; a.f();

Ответ:

1. 2 1

2. 1 2

3. 1

4. ошибка

Правильный ответ: 3

**Вопрос 62**

Если записано:

void f() { static int a; a = 4; }

int main() { int a = 9; f(); cout << a; return 0;}

то программа выведет следующее

Правильный ответ: 9

**Вопрос 63**

Если имеется код:

class A {

public:

A() { cout << 1; }

~A() { cout << 2; }

};

class B : public A {

public:

B() { cout << 3; }

~B() { cout << 4; }

};

void f(){

B b;

}

и создается объект класса B, то какой будет результат вызова функции f()?

Правильный ответ: 1342

**Вопрос 64**

Если создаются несколько экземпляров класса то:

Правильный ответ: они расположены в памяти программы

**Вопрос 65**

Жизненный цикл программного обеспечения (ПО) — период времени, который начинается с момента принятия решения о необходимости создания программного продукта и заканчивается в момент его полного изъятия из эксплуатации.

**Вопрос 66**

Идиома программирования — устойчивый способ выражения некоторой составной конструкции в одном или нескольких языках программирования. Идиома является шаблоном решения задачи, записи алгоритма или структуры данных путём комбинирования встроенных элементов языка.

Идиому можно считать самым низкоуровневым шаблоном проектирования, применяемым на стыке проектирования и кодирования на языке программирования. Идиома предписывает конкретный способ реализации определённых деталей и отношений между ними средствами конкретного языка. Набор идиом зависит от языка программирования, и одна и та же идиома может выглядеть в двух разных языках по-разному, либо в ней может не быть надобности в одном из них.

**Вопрос 67**

Известно, что в классе A определен один публичный конструктор A(int);. Выберите из предложенных выражений компилируемые:

1. A \*a(4);

2. A a;

3. A \*a = new A(4);

4. A a(4);

Правильный ответ: 3, 4

**Вопрос 68**

Какая из перечисленных функций не может быть конструктором String?

A. void String ()

B. String ();

C. String (String & s)

D. String (const int a)

Правильный ответ: А

**Вопрос 69**

Какая проблема в данной программе?

[font=courier]#include <iostream>

#include <cstring>

char \* CutString(const char \* str1)

{

char str2[4];

strncpy(str2, str1, 3);

str2[3] = ' '; //1

return str2;

}

int main()

{

char \* s = CutString("aaaaaaa");

std::cout<<(s);

} [/font]

Правильный ответ: должно быть str2[3] = '\0'; //1

**Вопрос 70**

Какие ключевые слова входят в состав инструкции "если":

1. if

2. then

3. Else

4. elseif

5. unless

Правильный ответ: 1

Пояснение: В C++ нет ключевого слова then. Кроме этого, C++ является регистрозависимым языком, поэтому ответ "Else" не является верным.

**Вопрос 71**

Какие строки вызовут ошибку компиляции?

#include <iostream>

template <typename T1, typename T2> void f(T1 x, T2 y) {

std::cout << x << " " << y << std::endl;

}

class A {

public:

A(int x) : val(x) { }

private:

int val;

};

int main() {

f<char>('1', "2"); // 1

f<string>('1', "2"); // 2

f('1', "2"); // 3

f<A, int> (1, 2); // 4

f(1.0, 2); // 5

return 0;

}

Правильный ответ: 2

**Вопрос 72**

Существуют два способа сообщить компилятору, что встраивание функций может быть желательно: с помощью ключевых слов **inline** и с помощью функций-членов класса.

**Вопрос 73**

По умолчанию все поля и методы класса являются **private**

**Вопрос 74**

Каковы значения целых x и y в выражении y=3+2\*(x=7/2);?

1. x=3, y=3

2. x=3.5, y=3

3. x=4, y=9

4. x=3, y=9

Правильный ответ: 4

**Вопрос 75**

В языке C++ **есть** 3 уровня доступа:

* спецификатор **public** делает члены открытыми;
* спецификатор **private** делает члены закрытыми;
* спецификатор **protected** открывает доступ к членам только для дружественных и дочерних классов (детально об этом на соответствующем уроке).

**Вопрос 76**

Какое значение будет содержать переменная y?

const int x = 5;

int main(int argc, char\*\* argv)

{

int x[x];

int y = sizeof(x) / sizeof(int);

return 0;

}

Правильный ответ: 5

**Вопрос 78**

Для генерации исключительной ситуации используется ключевое слово **catch**

**Вопрос 79**

Обработка исключений в С++ использует три ключевых слова: **try**, **catch** и **throw**. Те инструкции программы, где ожидается возможность появления исключительных ситуаций, содержатся в бло­ке **try**. Если в блоке **try** возникает исключение, т. е. ошибка, то генерируется исключение. Исключение перехватывается, используя **catch**, и обрабатывается. Инструкция **throw** должна выполняться либо внутри блока **try**, либо в функции, вызванной из блока **try**.

**Вопрос 80**

Какой будет результат следующей программы?

class t

{ public: int sum;

float f(int a)

{

sum++;

return sum \* a;}

} cl;

int main()

{

cl.sum = 10;

cout << cl.sum << " " << cl.f(4);

return 0;

}

Правильный ответ: 11 44

**Вопрос 81**

В определении класса используйте **friend** ключевое слово и имя функции, не являющейся членом, или другой класс, чтобы предоставить ему доступ к закрытым и защищенным членам класса. В определении шаблона параметр типа может быть объявлен дружественным.

class friend F

**Вопрос 82**

Какой конструктор вызывается при выполнении операции присваивания между существующими объектами одного типа?

1. конструктор по умолчанию.

2. конструктор копирования.

3. конструктор с параметром-объектом.

4. в этом случае никакой конструктор не вызывается.

Правильный ответ: 4

**Вопрос 83**

Какой модификатор доступа делает элементы объекта доступными в пределах иерархии классов?

Protected — доступ открыт классам, производным от данного. То есть, производные классы получают свободный доступ к таким свойствам или метода. Все другие классы такого доступа не имеют.

**Вопрос 84**

Какой модификатор доступа делает элементы объекта доступными всей программе?

Public — доступ открыт всем другим классам, кто видит определение данного класса.

**Вопрос 85**

Какой модификатор доступа делает элементы объекта доступными только методам данного класса?

Private — доступ открыт самому классу (т.е. функциям-членам данного класса) и друзьям (friend) данного класса - как функциям, так и классам. Однако производные классы не получают доступа к этим данным совсем. И все другие классы такого доступа не имеют.

**Вопрос 86**

Какой оператор в C++ используется для взятия адреса?

Оператор & можно запомнить как «взятие адреса».

**Вопрос 87**

Какой правильный вызов функции базового класса из объекта производного класса, если в производном классе эта функция была замещена?

1. FunctionName();

2. Base::FunctionName();

3. Base.FunctionName();

4. такую функцию вызывать нельзя.

Правильный ответ: 2

**Вопрос 88**

Какой результат будет у следующего выражения?

class A

{ public : int a, b;

protected : int z;

private : short i; } A1;

class B : private A

{ private : int k;} B1;

int main()

{ B1.a = 10;

cout << B1.a;

}

Ответ:

1. 10

2. ошибка

3. 0

Правильный ответ: 2

**Вопрос 89**

Какой результат следующего выражения ?

int\* a; int b; a = &b; b = 7; \*a++; cout << b;

1. 7

2. 8

3. не определено

4. ошибка компиляции

Правильный ответ: 2

**Вопрос 90**

Какой уровень доступа имеет переменная "x" из класса "derived" в приведенном примере? class basex { protected: int x; }; class derived : private basex {};

1. protected

2. global

3. private

4. public

5. local

Правильный ответ: 3

**Вопрос 91**

Какой шаблон STL используется для создания динамических массивов вида "ключ-значение"?

**set** - это коллекция уникальных значений const Key - каждое из которых является также и ключом - то есть, проще говоря, это отсортированная коллекция, предназначенная для быстрого поиска необходимого значения. К ключу предъявляются те же требования, что и в случае ключа для map. Естественно, использовать ее для этой цели нет смысла, если вы хотите сохранить в ней простые типы данных, по меньшей мере вам необходимо определить свой класс, хранящий пару ключ - значение и определяющий операцию сравнения по ключу. Очень удобно использовать данную коллекцию, если вы хотите избежать повторного сохранения одного и того же значения. Для того, чтобы начать использование данной коллекции, включите **#include <set>**

**Вопрос 92**

Какой шаблон STL используется для создания массивов в стиле С?

**vector** - коллекция элементов Т, сохраненных в массиве, увеличиваемом по мере необходимости. Для того, чтобы начать использование данной коллекции, включите #include <vector>.

**Вопрос 93**

Какой шаблон STL оптимален для создания двунаправленной очереди?

**list** - коллекция элементов Т, сохраненных, как двунаправленный связанный список. Для того, чтобы начать использование данной коллекции, включите #include <list>.

**Вопрос 94**

Какую величину вычисляет программа?

#include <stdio.h>

main()

{ int a[3][4];

int i, j, k;

for (i = 0; i < 3; i++)

for (j = 0; j < 4; j++) scanf("%d", &a[i][j]);

k = -1;

for (i = 0; i < 4; i++) {

for (j = 0; j < 3; j++)

if (a[j][i] == 0) k = i;

}

printf("%d",k);

}

1. номер первого из столбцов массива, содержащих хотя бы один элемент, равный 0
2. номер последнего из столбцов массива, содержащих хотя бы один элемент, равный 0
3. количество положительных элементов массива в каждом столбце
4. номер первой из строк массива, содержащих хотя бы один элемент, равный 0
5. номер последней из строк массива, содержащих хотя бы один элемент, равный 0

Правильный ответ: 2

**Вопрос 95**

Какую величину вычисляет программа?

#include <stdio.h>

main(){

int a[3][4];

int i, j, k;

for (i = 0; i < 3; i++)

for (j = 0; j < 4; j++) scanf("%d",&a[i][j]);

for (i = 0; i < 3; i++)

{k = 0; for (j = 0; j < 4; j++)

if (a[i][j] > 0) ++k;

}

printf("%d",k);

}

1. количество положительных элементов массива

2. количество положительных элементов массива в каждой строке

3. количество положительных элементов массива в последней строке

4. количество положительных элементов массива в последнем столбце

Правильный ответ: 3

**Вопрос 96**

Класс — в объектно-ориентированном программировании, представляет собой шаблон для создания объектов, обеспечивающий начальные значения состояний: инициализация полей-переменных и реализация поведения функций или методов. Другие абстрактные типы данных — метаклассы, интерфейсы, структуры, перечисления, — характеризуются какими-то своими, другими особенностями. Класс является ключевым понятием в ООП. Существуют и бесклассовые объектно-ориентированные языки, например, Self, Lua, (подробнее смотрите Прототипное программирование). Суть отличия классов от других абстрактных типов данных состоит в том, что при задании типа данных класс определяет одновременно как интерфейс, так и реализацию для всех своих экземпляров (т.е. объектов), вызов метода-конструктора обязателен.

**Вопрос 97**

**Коммит**. Точно так же, как и в игре, в системе контроля версий Git можно сохранить текущее состояние проекта. Для этого есть специальная команда — commit. Она делает так, что новая версия проекта сохраняется и добавляется в хранилище. В файле с сохранением отображаются: все изменения, которые происходили в рабочей области, автор изменений и краткий комментарий, описывающий суть изменений. Каждый коммит хранит полное состояние рабочей области, её папок и файлов проекта.

В итоге проект работает так:

* Репозиторий хранит все версии проекта. В случае передачи этого проекта другому человеку, он увидит всё, что с ним происходило до этого.
* Ничего не теряется и не удаляется бесследно. При удалении файла в новой версии добавляется запись о том, что файл был удалён.
* Всегда можно вернуться к любой из версий проекта, загрузив её из хранилища в рабочую область.

**Вопрос 98**

Компилируется ли следующий код:

#include <iostream>

using namespace std;

class A {

A () {}

~A () {}

public:

void dd () { cout << "DDD"; }

};

int main () {

A \*a = new A ();

a->dd ();

delete a;

return 0;

}

Правильный ответ: код не компилируется

**Вопрос 99**

Могут ли управляющие конструкции быть вложенными?

Операторы выбора, как и любые управляющие конструкции, могут быть вложенными, чтобы результат одного ответа помогал определить, как ответить на следующий.

**Вопрос 100**

Наблюдатель (англ. Observer) — поведенческий шаблон проектирования. Также известен как «подчинённые» (англ. Dependents). Реализует у класса механизм, который позволяет объекту этого класса получать оповещения об изменении состояния других объектов и тем самым наблюдать за ними.

Классы, на события которых другие классы подписываются, называются субъектами (Subjects), а подписывающиеся классы называются наблюдателями (англ. Observers).

**Вопрос 101**

Найдите недопустимую запись вещественной константы:

1. 12.3e+2

2. 1.23

3. .1f

4. 0x3F

Правильный ответ: 4

**Вопрос 102**

Объе́кт в программировании — некоторая сущность в цифровом пространстве, обладающая определённым состоянием и поведением, имеющая определённые свойства (атрибуты) и операции над ними (методы). Как правило, при рассмотрении объектов выделяется то, что объекты принадлежат одному или нескольким классам, которые определяют поведение (являются моделью) объекта. Термины «экземпляр класса» и «объект» взаимозаменяемы.

**Вопрос 103**

Объявляется класс T. Какой из следующих прототипов конструкторов является конструктором копирования?

1. T();
2. T(int, int);
3. T(const T&);

Правильный ответ: 3

**Вопрос 104**

Объявляется класс T. Какой из следующих прототипов конструкторов является конструктором по умолчанию?

1. T();
2. T(int, int);
3. T(const A&);

Правильный ответ: 1

**Вопрос 105**

Одиночка — это порождающий паттерн проектирования, который гарантирует, что у класса есть только один экземпляр, и предоставляет к нему глобальную точку доступа.

**Вопрос 106**

Определить, какие выражения ложны:

'a' > 'c' 100>3 && 'a'>'c'

!(100 > 3) !(101 >= 99)

1. все выражения ложны

2. ложно только !(100>3)

3. ложно только !(101>=99)

4. все выражения истинны

Правильный ответ: 1

**Вопрос 108**

Отметьте правильное создание объекта вложенного класса cls2, если имеется следующее объявление:

class cls1 { public : int yy; class cls2 { public : int zz; }; };

1. cls2 c2;
2. cls1::cls2 c2;

Правильный ответ: 2

**Вопрос 109**

Отметьте правильное утверждение для абстрактного класса для языка С ++.

A. Класс, у которого все методы чисто виртуальные, называется абстрактным.

B. Абстрактный базовый класс навязывает определенный интерфейс всем производным из него классам.

C. Невозможно создать объект абстрактного класса.

D. В абстрактном классе не описываются методы вообще.

Правильный ответ: С

**Вопрос 110**

Паттерны (или шаблоны) проектирования описывают типичные способы решения часто встречающихся проблем при проектировании программ.

**Вопрос 111**

const Integer operator+(const Integer& rv) const { return (value + rv.value);

**Вопрос 112, 113**

Конструкторы и деструкторы при наследовании

Если и у базового и у производного классов есть конструкторы и деструкторы, то конструкторы выполняются в порядке наследования, а деструкторы – в обратном порядке.

Следует запомнить, что конструкторы вызываются в том порядке, в котором классы выводились один из другого. Порядок вызова деструкторов обратный.

**Вопрос 114**

Поведенческие шаблоны (англ. behavioral patterns) — шаблоны проектирования, определяющие алгоритмы и способы реализации взаимодействия различных объектов и классов.

В поведенческих шаблонах уровня класса используется наследование, чтобы определить поведение для различных классов. В поведенческих шаблонах уровня объекта используется композиция. Некоторые из них описывают, как с помощью кооперации несколько равноправных объектов работают над заданием, которое они не могут выполнить по отдельности. Здесь важно то, как объекты получают информацию о существовании друг друга. Объекты-коллеги могут хранить ссылки друг на друга, но это усиливает степень связанности системы. При высокой связанности каждому объекту пришлось бы иметь информацию обо всех остальных. Некоторые из шаблонов решают эту проблему.

**Вопрос 115**

Полиморфизм в языках программирования и теории типов — способность функции обрабатывать данные разных типов

**Вопрос 116**

Каждый метод имеет в качестве параметра указатель \*this, который указывает на адрес объекта, с которым в данный момент выполняется операция.

**Вопрос 117**

Порождающие шаблоны — шаблоны проектирования, которые имеют дело с процессом создания объектов. Они позволяют сделать систему независимой от способа создания, композиции и представления объектов. Шаблон, порождающий классы, использует наследование, чтобы изменять наследуемый класс, а шаблон, порождающий объекты, делегирует инстанцирование другому объекту.

**Вопрос 118**

Правило трёх (aka «Закон Большой Тройки», «Большая Тройка», "The Rule of Three") — правило в C++, гласящее, что если класс или структура определяет один из следующих методов, то они должны явным образом определить все три метода.

О каких методах идёт речь (выберите из списка)?

1. Оператор = (присваивания копированием)
2. Оператор
3. Оператор >
4. Конструктор
5. Конструктор копирования
6. Деструктор

Правильный ответ: 1, 5, 6

**Вопрос 119**

Система управления версиями — программное обеспечение для облегчения работы с изменяющейся информацией. Система управления версиями позволяет хранить несколько версий одного и того же документа, при необходимости возвращаться к более ранним версиям, определять, кто и когда сделал то или иное изменение, и многое другое.

Часто выполнить слияние невозможно ни в автоматическом, ни в ручном режиме, например, если формат файла неизвестен или слишком сложен. Некоторые системы управления версиями дают возможность заблокировать файл в хранилище. Блокировка не позволяет другим пользователям получить рабочую копию или препятствует изменению рабочей копии файла (например, средствами файловой системы) и обеспечивает, таким образом, исключительный доступ только тому пользователю, который работает с документом.

**Вопрос 120**

Ревизия характеризует состояние не отдельного файла, а всего хранилища в целом. Например, ревизия 32 (обведено пунктиром на рисунке) — это состояние четырёх файлов и двух каталогов, существовавших в хранилище на тот момент.

Номер ревизии является аналогом времени в том смысле, что меньшие номера ревизий соответствуют более ранним состояниям хранилища, а бо́льшие — поздним.

**Вопрос 121**

Репозито́рий — место, где хранятся и поддерживаются какие-либо данные. Чаще всего данные в репозитории хранятся в виде файлов, доступных для дальнейшего распространения по сети.

Репозитории используются в системах управления версиями, в них хранятся все документы вместе с историей их изменения и другой служебной информацией.

**Вопрос 122**

Сборка мусора в программировании — одна из форм автоматического управления памятью. Специальный процесс, называемый сборщиком мусора, периодически освобождает память, удаляя объекты, которые уже не будут востребованы приложениями.

**Вопрос 123**

Сущности – это абстракции, которые являются основными элементами модели, **связи** соединяют их между собой, а диаграммы группируют представляющие интерес наборы сущностей.

**Вопрос 124**

Система управления версиями — программное обеспечение для облегчения работы с изменяющейся информацией. Система управления версиями позволяет хранить несколько версий одного и того же документа, при необходимости возвращаться к более ранним версиям, определять, кто и когда сделал то или иное изменение, и многое другое.

**Вопрос 125**

Системы управления версиями бывают:

* Локальные
* Централизованные
* Распределенные

**Вопрос 126**

Структурные шаблоны — шаблоны проектирования, в которых рассматривается вопрос о том, как из классов и объектов образуются более крупные структуры.

Структурные шаблоны уровня класса используют наследование для составления композиций из интерфейсов и реализаций. Простой пример — использование множественного наследования для объединения нескольких классов в один. В результате получается класс, обладающий свойствами всех своих родителей. Особенно полезен этот шаблон, когда нужно организовать совместную работу нескольких независимо разработанных библиотек.

**Вопрос 127**

Перехват всех исключений catch(…)

Предложение catch(...) используется самостоятельно или в сочетании с другими catch-обработчиками. В последнем случае следует позаботиться о правильной организации обработчиков, ассоциированных с try-блоком.

**Вопрос 128**

Чему равно значение выражения (a || b && a || c), где a, b ис - величины типа bool, имеющие значения false, true и true соответственно?

1. false

2. true

Правильный ответ: 2

**Вопрос 129**

Чему равно числовое значение выражения sqrt(4)+142/20\*2?

1. 12

2. 16

3. 5

Правильный ответ: 2

**Вопрос 130**

Члены класса - это переменные состояния и методы этого класса, иными словами членами класса могут быть как переменные, так и функции. Функции и переменные, объявленные внутри объявления класса, становятся членами этого класса. ... По умолчанию, все функции и переменные, объявленные в классе, становятся закрытыми (private).

**Вопрос 131**

Что будет выведено на экран?

#include <iostream>

class Base

{

public:

Base() { PrintClassName(); }

virtual void PrintClassName() { std::cout<<"Base"; }

};

class Derived: public Base

{

public:

Derived() { }

void PrintClassName() { std::cout<<"Derived"; }

};

int main(int argc, char \*argv[])

{

Derived d;

}

Правильный ответ: Base

**Вопрос 132**

Что будет на экране?

#include <iostream>

using namespace std;

int main()

{

cout << "Hello, World!" + 3 << endl;

return 0;

}

Правильный ответ: lo, World!

**Вопрос 133**

Что будет напечатано в консоль?

#include <iostream>

int main()

{

int \* var = new int(2);

\*var \*= 2;

std::cout << \*var;

return 0;

}

Правильный ответ: 4

**Вопрос 134**

Что будет напечатано в консоль?

#include <iostream>

int main()

{

int a = 1 << 2;

switch (a)

{

default:

std::cout << "0";

case 1:

std::cout << "1"; ++a; break;

case 2:

std::cout << "2";

case 3:

std::cout << "3"; break;

}

return 0;

}

Правильный ответ: 01

**Вопрос 135**

Что будет результатом следующей программы?

#include <iostream>

using namespace std;

class A {

public:

A();

friend A &operator+(A &, const int);

int getValue() const;

private:

int val;

};

A::A() { val = 0; }

int A::getValue() const { return val; }

A &operator+ (A &sa, const int a) {

sa.val += a;

return sa;

}

int main(void) {

A q, w;

w + 3; // 1

q = q + 2; // 2

w = w + q.getValue(); // 3

cout << q.getValue() << w.getValue();

return 0;

}

Правильный ответ: 25

**Вопрос 136**

Что выведется на экран в результате выполнения следующего фрагмента кода:

int i = 2;

switch (i) {

case 0: cout << 0;

case 1: cout << 1;

case 2:

case 3: cout << 3;

default: cout << 4;

}

Правильный ответ: 34

**Вопрос 137**

Что выведется на экран в результате работы следующего кода?

class Base

{

public:

Base(int \_x = 10){x = \_x;}

void f() {cout<<x\*2<<endl;}

protected:

int x;

};

class Derived: public Base

{

public:

void f() {cout<<x/2<<endl;}

};

int main()

{

Derived d;

Base\* pB= &d;

pB->f();

return 0;

}

Правильный ответ: 20

**Вопрос 138**

При разработке было добавлено ключевое слово inline , которое применяется к функциям. Обычно это означает, что при компиляции код этой функции будет вставляться на месте вызовов. ... Однако ключевое слово inline является всего лишь запросом к компилятору и может быть проигнорировано.

**Вопрос 140**

Что напечатает следующий код:

#include<iostream>

struct A {

~A() {

std::cout << "A";

}

};

struct B : public A {

B() {

throw 1;

}

~B() {

std::cout << "B";}};

int main() {

try {

B b;}

catch(...) { }}

Правильный ответ: А

**Вопрос 141**

Что напечатает следующий код:

#include <iostream>

using namespace std;

void print(int x, int y) {

cout << x << y << endl; }

int main() {

int x = 1;

int y = 2;

print(y = 5, x = 6);}

Правильный ответ: 56

**Вопрос 142**

Использование древовидной структуры означает, что каждый объект типа QObject может иметь одного родителя и ноль или несколько детей. Родительский объект управляет временем жизни дочерних объектов.

**Вопрос 143**

В языках программирования единица трансляции — максимальный блок исходного текста, который физически можно оттранслировать (преобразовать во внутреннее машинное представление; в частности, откомпилировать).

**Вопрос 144**

Значение ложь (false) 0; // это ложь

false; // это тоже ложь, причем это слово есть только в С++

В языках Си и С++ значение 0 и его эквиваленты: NULL или nullptr являются значениями ложи.

**Вопрос 145**

Значение истины (true)

Любое целочисленное значение, кроме 0 будет эквивалентно истине. Например, 1, 2, 4, 45 — все это истина, а нуль — нет. В С++ также зарезервировано ключевое слово, обозначающее истину — true.