Пример теста для самоконтроля.

*Используется 3 типа заданий:*

1. *С выбором ответа: На каждый вопрос дано по 4 ответа: А, В, С, Д. Обведите один правильный ответ.*
2. *Написать определение данному понятию*
3. *Вставить команды*

1. С помощью чего реализуется принцип полиморфизма в С++?
   1. наличия множественного наследования. B. наличия виртуальных методов.

C. использование виртуального наследования. D. наличия абстрактных классов.

1. В программе описаны класс и указатель на объект:

class A {public: int a, b, c; };

…

* 1. \* obj = new A;

Как обратиться к полю c?

A. obj.c B. obj-> **с** C. obj A -> -> с D. obj-> A.с

1. Какая из перечисленных функций не может быть конструктором?
   1. void String ()B. String (); C. String (String & s) D. String (const int a)
2. Отметьте правильное утверждение для абстрактного класса для языка С++.
   1. Класс, у которого все методы чисто-виртуальные, называется абстрактным.
   2. Абстрактный базовый класс навязывает определенный интерфейс всем производным из него классам.
   3. Невозможно создать объект абстрактного класса.
   4. В абстрактном классе не описываются методы вообще.
3. Даны 2 фрагмента программы и 2 возможных результата их работы.

**Ф1.** class A { protected: int x;

public: A(){cout<< "Constr-r A default \n"; };

A(int i){cout<< "Constr-r A initialized \n"; }; };

class B { protected: int y; public: void Show(void) {cout<< "function Show \n"; };

};

int main() { A aobj(25); A a1obj; B b1obj; b1obj.Show(); return 0; }

**Ф2.** class A { protected: int x;

public: A(){cout<< "Constr-r A default \n"; };

A(int i){cout<< "Constr-r A initialized \n"; }; }; class B: public A { protected: int y; public: void Show(void) {cout<< "function Show \n"; };

};

int main() { A aobj(25); A a1obj; B b1obj; b1obj.Show(); return 0; }

|  |  |
| --- | --- |
| P1. | P2. |

Найдите верное соответствие фрагмента и результата

**А.** Ф1– Р1 Ф2– Р2 **В.** Ф1– Р2 Ф2 – Р1 **С.** Ф1 – Р1 Ф2 – нет ответа **D.**  Ф1– Р2 Ф2 – нет ответа

1. Какой из перечисленных методов может быть конструктором для класса TString в языке С++?
   1. TString \* TString (); B. void TString (); C. TString (int х); D. const TString (int a);
2. Какая функция, не являющаяся методом класса, имеет доступ к его защищенным элементам? A. Шаблонная. B. Полиморфная. C. Дружественная. D. Виртуальная.
3. Вызовет ли данный код ошибку компиляции?

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

A. Ошибки нет, все записано верно.

* 1. Ошибка: имя деструктора должно совпадать с именем класса.
  2. Ошибка: имя деструктора не может начинаться с маленькой буквы.
  3. Ошибка: никакой из идентификаторов в С ++ не может начинаться со знака «~».

1. Укажите правильное объявление виртуального метода в С++, который принимает одно целочисленное значение и возвращает void.
   1. virtual void SomeFunction (int х);
   2. void SomeFunction (int х) virtual;
   3. virtual SomeFunction (int х);
   4. virtual void SomeFunction (int \* x);
2. Есть функция: void f (int i=0; int j=1) {}; Впишите варианты вызова функции f и значения i и j:

Вызов: f() i =0 j =1 Вызов: f(6) i = 6 j = 1 Вызов: f(10,3) i = 10 j = 3

1. Назовите принцип объектно-ориентированного программирования, который заключается в объединении полей и методов объекта с целью обеспечения сохранности данных.
   1. Наследование. B. Сочетание. C. Инициализация.D. Инкапсуляция.
2. В программе описаны абстрактный класс А и производный от этого класса - класс А1. фКакой фрагмент кода заведомо неверный?
   1. A \* a = new A; B. A1 a1; C. A a1; A1 a2;D. A1 a1; A1 a2;
3. Выберите верное утверждение о деструкторе класса в С ++.
   1. Деструктор принимает в качестве параметра адрес того объекта, который нужно уничтожить.
   2. Деструктор принимает в качестве параметра указатель this.
   3. Деструктор не содержит параметров.
   4. Деструктор принимает в качестве параметра тот объект, который нужно уничтожить.
4. Укажите правильный вариант доступа к элементам класса (язык С ++), описанных следующим образом:

class my { char s; public: double Z; int f (int c, int d) {return c + d;}; } T1, T2;

A. T1.Z = 23.1;B. T2-> f (2,1); C. T1.s = ‘#’; D. my.T2-> s = ‘L‘;

1. Какой из вариантов записи абстрактного класса в С++ является правильным?
   1. abstract class A {virtual int f () = 0;}; B. class A {virtual int f () = 0;};

C. class A {virtual int f () = 0;} abstract; D. class A {virtual int f ();};

1. Что будет выведено на экран по окончанию программы?

class A { public: A(){this->\_num = 0;}

A(int num){this->\_num = num; cout<<this->\_num;}

~A(){cout<<this->\_num;} private: int \_num; };

int main(void){ A val(100); return 0; }

* 1. Ничего выведено не будет, т. к. поле “int \_num“ скрыто от конструктора;
  2. 100; C. 100100;

D. Ничего выведено не будет, т. к. имя поля не должно начинаться с нижнего подчёркивания;

1. Каким является модификатор доступа к элементу класса по умолчанию?
   1. public; B. private;C. protected; D. static;
2. Дана программа:

|  |  |
| --- | --- |
| class X  { protected: int x;  public:  X():x(0){cout<<" X1: "  <<" x="<<x<<endl;}  X(int i):x(i){cout<<"  X2: "<<" x="<<x<<endl;}  };  class Y: public X { public: int y;}; int main()  {X obj\_x(5); Y obj\_y;} | Что выведется на экран после ее работы:  X2: x=5 X1: x=0    \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

1. Что понимается под определением в одной области действия нескольких функций с одинаковым именем, но различными параметрами?
   1. Переопределение; B. Дружественность; C. Перегрузка;D. Наследование;
2. Что будет выведено на экран по окончанию программы?

class user {private: int id; static int next\_id; public: static int next\_user\_id(){next\_id++; return next\_id;} user(){id = user::next\_id++;}

}; int user::next\_id = 0; int main() {user obj; user obj2; cout<<obj2.next\_user\_id(); return 0;}

* 1. 0; B. 1; C. 2; D. 3;

21.Что напечатает следующий код при создании объекта (экземпляра) класса X:

class Y { public: Y() { cout<<"Y"; } }; class Z { public: Z() { cout<<"Z"; } }; class X : public Z { private: Y m\_objY; public: X(){ cout<<"X"; } }; int main() {X obj; return 0;}

A. ZX; B. YX; C. ZYX;D. X;

1. Есть класс

**class MyClass** { private: int x;

public: void f1(int d) { d = x; } }; В следующем фрагменте замените фразы на команды:

int main() { Myclass obj; создание объекта класса; obj.f1(10); вызов метода f1;

return 0;}

1. Чем обличается класс от объекта?\_\_\_\_\_ класс это шаблон, а объект строится по этому шаблону, и с ним мы уже работаем \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_
2. Что такое метод?\_\_ это [функция](https://ru.wikipedia.org/wiki/%D0%A4%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D1%8F_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)) или [процедура](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%86%D0%B5%D0%B4%D1%83%D1%80%D0%B0_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)), принадлежащая какому-то [классу](https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B0%D1%81%D1%81_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)) или [объекту](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)).
3. Может ли метод быть приватный?\_да\_\_\_\_\_
4. Можно в методах присваивать параметрам значения по умолчанию?\_\_\_да\_\_\_\_
5. Чем отличается модификатор доступа protected:

от модификатора private\_\_\_\_\_\_ доступ открыт только внутри данного класса, а protected открыт также для наслед. классов \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

от модификатора public\_\_\_\_\_ доступ открыт абсолютно всем, кто видит определение данного класса \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Укажите правильное объявление виртуального метода в С++, который принимает одно целочисленное значение и возвращает void.
   1. virtual void SomeFunction (int х); B. void SomeFunction (int х) virtual;

C. virtual SomeFunction (int х); D. virtual void SomeFunction (int \* x);

1. Как нужно реализовать статическую функцию «next\_user\_id», чтобы выполнился следующий код (функция должна возвращать значение id следующего от текущего объекта!!!):

#include <iostream>

class user

{ private: int id; static int next\_id;

public: static int next\_user\_id(){…} user(){id = user::next\_id++;} int getId(){return id;}

};

int user::next\_id = 0; int main() {user obj; … return 0;}

* 1. static int next\_user\_id(){return Id+1;};
  2. static int next\_user\_id(){return next\_id;};
  3. static int next\_user\_id(user obj){return obj.id+1;};
  4. static int next\_user\_id(user obj){return obj.getId()+1;};