

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 C++?

- А. наличия множественного наследования.      В. наличия виртуальных методов.  
С. использование виртуального наследования.      Д. наличия абстрактных классов.

2. В программе описаны класс и указатель на объект:

```
class A {public: int a, b, c; };  
...  
A * obj = new A;
```

Как обратиться к полю c?

- А. obj.c                      В. obj->c      С. obj A -> -> c                      Д. obj-> A.c

3. Какая из перечисленных функций не может быть конструктором?

- А. void String ()              В. String ();      С. String (String & s)      Д. String (const int a)

4. Отметьте правильное утверждение для абстрактного класса для языка C++.

- А. Класс, у которого все методы чисто-виртуальные, называется абстрактным.  
В. Абстрактный базовый класс навязывает определенный интерфейс всем производным из него классам.  
С. Невозможно создать объект абстрактного класса.  
Д. В абстрактном классе не описываются методы вообще.

5. Даны 2 фрагмента программы и 2 возможных результата их работы.

❖1. 

```
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 alobj; B blobj; blobj.Show(); return 0; }
```

❖2. 

```
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 alobj; B blobj; blobj.Show(); return 0; }
```

P1.

```
C:\ C:\DOCUME~1\sitn\LOCALS~1
Constr-r A initialized
Constr-r A default
Constr-r A default
function Show
```

P2.

```
C:\ C:\DOCUME~1\sitn\LOCALS~1
Constr-r A initialized
Constr-r A default
function Show
```

Найдите верное соответствие фрагмента и результата

A. Ф1– P1 Ф2– P2 B. Ф1– P2 Ф2 – P1 C. Ф1 – P1 Ф2 – нет ответа D. Ф1– P2 Ф2 – нет ответа

6. Какой из перечисленных методов может быть конструктором для класса TString в языке C++?

A. TString \* TString (); B. void TString (); C. TString (int x); D. const TString (int a);

7. Какая функция, не являющаяся методом класса, имеет доступ к его защищенным элементам?

A. Шаблонная. B. Полиморфная. C. Дружественная. D. Виртуальная.

8. Вызовет ли данный код ошибку компиляции?

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

A. Ошибки нет, все записано верно.

B. Ошибка: имя деструктора должно совпадать с именем класса.

C. Ошибка: имя деструктора не может начинаться с маленькой буквы.

D. Ошибка: никакой из идентификаторов в C ++ не может начинаться со знака «~».

9. Укажите правильное объявление виртуального метода в C++, который принимает одно целочисленное значение и возвращает void.

A. virtual void SomeFunction (int x);

B. void SomeFunction (int x) virtual;

C. virtual SomeFunction (int x);

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

10. Есть функция: void f (int i=0; int j=1) {}; Впишите варианты вызова функции f и значения i и j:

Вызов: \_\_\_\_\_ i = \_\_\_\_\_ j = \_\_\_\_\_ Вызов: \_\_\_\_\_ i = \_\_\_\_\_ j = \_\_\_\_\_ Вызов: \_\_\_\_\_ i = \_\_\_\_\_ j = \_\_\_\_\_

11. Назовите принцип объектно-ориентированного программирования, который заключается в объединении полей и методов объекта с целью обеспечения сохранности данных.

A. Наследование. B. Сочетание. C. Инициализация. D. Инкапсуляция.

12. В программе описаны абстрактный класс A и производный от этого класса - класс A1. Какой фрагмент кода заведомо неверный?

A. A \* a = new A; B. A1 a1; C. A a1; A1 a2; D. A1 a1; A1 a2;

13. Выберите верное утверждение о деструкторе класса в C++.

- A. Деструктор принимает в качестве параметра адрес того объекта, который нужно уничтожить.
- B. Деструктор принимает в качестве параметра указатель this.
- C. Деструктор не содержит параметров.
- D. Деструктор принимает в качестве параметра тот объект, который нужно уничтожить.

14. Укажите правильный вариант доступа к элементам класса (язык C++), 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';

15. Какой из вариантов записи абстрактного класса в C++ является правильным?

- A. 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 ();};

16. Что будет выведено на экран по окончании 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; }
```

- A. Ничего выведено не будет, т. к. поле "int \_num" скрыто от конструктора;
- B. 100;    C. 100100;
- D. Ничего выведено не будет, т. к. имя поля не должно начинаться с нижнего подчёркивания;

17. Каким является модификатор доступа к элементу класса по умолчанию?

- A. public;    B. private;    C. protected;    D. static;

18. Дана программа:

<pre>class X { protected: int x;   public:     X():x(0){cout&lt;&lt;" X1: "&lt;&lt;" x="&lt;&lt;x&lt;&lt;endl;}     X(int i):x(i){cout&lt;&lt;" X2: "&lt;&lt;" x="&lt;&lt;x&lt;&lt;endl;} }; class Y: public X { public: int y;}; int main() {X obj_x(5); Y obj_y;}</pre>	<p>Что выведется на экран после ее работы:</p> <hr/> <hr/> <hr/>
---	--

19. Что понимается под определением в одной области действия нескольких функций с одинаковым именем, но различными параметрами?

A. Переопределение; B. Дружественность; C. Перегрузка; D. Наследование;

20. 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;}
```

A. 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;

22. Есть класс

```
class MyClass {
private:    int x;
public:    void f1(int d) { d = x; } };
```

В следующем фрагменте замените фразы на команды:

```
int main() {
    создание объекта класса; _____
    вызов метода f1; _____
    return 0;}
```

23. Чем обличается класс от объекта? \_\_\_\_\_

—

24. Что такое метод? \_\_\_\_\_

25. Может ли метод быть приватный? \_\_\_\_\_

26. Можно в методах присваивать параметрам значения по умолчанию? \_\_\_\_\_

27. Чем отличается модификатор доступа protected:

от модификатора private \_\_\_\_\_

от модификатора public \_\_\_\_\_

28. Укажите правильное объявление виртуального метода в C++, который принимает одно целочисленное значение и возвращает void.

- A. virtual void SomeFunction (int x);    B. void SomeFunction (int x) virtual;  
C. virtual SomeFunction (int x);        D. virtual void SomeFunction (int \* x);

29. Как нужно реализовать статическую функцию «next\_user\_id», чтобы выполнялся 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;}
```

- A. static int next\_user\_id(){return Id+1;};  
B. static int next\_user\_id(){return next\_id;};  
C. static int next\_user\_id(user obj){return obj.id+1;};  
D. static int next\_user\_id(user obj){return obj.getId()+1;};