

Объектно-ориентированное программирование  
В вопросах о языке программирования используется C++  
(стандарт C++17). Все задания предполагают выбор одного  
или нескольких правильных ответов из предложенных.

### 1. Задание. Согласно парадигме ООП...

- ☐ Программа должна иметь графический интерфейс пользователя
- ☐ Программ должна строиться из: оператора последовательности, условного оператора и оператора цикла
- ☐ Программ должна строиться из модулей
- ☒ В основе программы должны лежать классы
- ☐ Программа строится из классов стандартной библиотеки

### 2. Задание. Основные принципы ООП

- |   |  |
|---|--|
| <input checked="" type="checkbox"/> Абстрагирование | <input checked="" type="checkbox"/> Полиморфизм  |
| <input checked="" type="checkbox"/> Наследование    | <input type="checkbox"/> Модульность             |
| <input type="checkbox"/> Агрегация                  | <input checked="" type="checkbox"/> Инкапсуляция |
| <input type="checkbox"/> Декомпозиция               | <input type="checkbox"/> Классификация           |

### 3. Задание

```
1. class Example{
2.     private:
3.         int y;
4.         float fun1();
5.     public:
6.         float x, w;
7.     public:
8.         void fun2();
9. };
10.
11. int main(){
12.     Example w, z1;
13.     int y;
14. }
```

Выберите правильные утверждения

- ☒ Код синтаксически верен
- ☐ Программа не компилируется: конфликт имён в строках 12 и 6
- ☐ Программа не компилируется: конфликт имён в строках 12 и 6, 13 и 3.
- ☐ Программа не компилируется: конфликту имён в строках 13 и 3.

*Миссерово Анна, ВМК-20*

### 4. Задание

```
1. class T{
2.     private:
3.         int a;
4.         float bar();
5.     public:
6.         float b, c;
7.     public:
8.         void foo();
9. };
10.
11. void baz(int a){
12.     a = a + 1; }
13.
14. int main(){
15.     T c, c1;
16.     int a1;
17. }
```

Полями класса являются...

- |  |                               |
|--|-------------------------------|
| <input checked="" type="checkbox"/> a (строка 3) | <input type="checkbox"/> foo  |
| <input checked="" type="checkbox"/> b (строка 6) | <input type="checkbox"/> bar  |
| <input checked="" type="checkbox"/> c (строка 6) | <input type="checkbox"/> baz  |
| <input type="checkbox"/> a (строка 11)           | <input type="checkbox"/> main |
| <input type="checkbox"/> c, c1 (строка 15)       | <input type="checkbox"/> T    |

### 5. Задание

```
1. class T{
2.     private:
3.         int a;
4.         float bar();
5.     public:
6.         float b, c;
7.     public:
8.         void foo();
9. };
10.
11. void baz(int a){
12.     a = a + 1; }
13.
14. int main(){
15.     T c, c1;
16.     int a1;
17. }
```

Методами класса T являются...

- |  |  |
|--|--|
| <input checked="" type="checkbox"/> bar (строка 4) | <input type="checkbox"/> a (строка 3)      |
| <input checked="" type="checkbox"/> foo (строка 8) | <input type="checkbox"/> b                 |
| <input type="checkbox"/> baz                       | <input type="checkbox"/> c (строка 6)      |
| <input type="checkbox"/> main                      | <input type="checkbox"/> a (строка 11)     |
| <input type="checkbox"/> T                         | <input type="checkbox"/> c, c1 (строка 15) |

### 6. Задание

```
class T{
private:
    int a;
    float bar();
public:
    float b, c;
public:
    void foo();
};

int main(){
    T c, *c1;
}
```

Следующие обращения к членам класса ошибочны...

- |  |                                 |
|--|---------------------------------|
| <input type="radio"/> c.a; <del>private</del>  | <input type="radio"/> c->bar;   |
| <input type="radio"/> c->a; <del>private</del> | <input type="radio"/> c.foo();  |
| <input type="radio"/> c.bar;                   | <input type="radio"/> c->foo(); |
| <input type="radio"/> c.bar();                 | <input type="radio"/> c.main(); |
| <input type="radio"/> c->bar();                | <input type="radio"/> c.main    |
|  | <del>c-&gt;foo</del>            |

### 7. Задание

```
1. class T{
2. private:
3.     int a;
4.     float bar();
5. public:
6.     float b, c;
7. public:
8.     void foo();
9. };
10.
11. void baz(int a){
12.     a = a + 1; }
13.
14. int main(){
15.     T c, c1;
16.     int a1;
17. }
```

Объектами являются...

- |   |                                      |
|---|--------------------------------------|
| <input type="radio"/> a                         | <input type="radio"/> baz            |
| <input type="radio"/> a (строка 11)             | <input type="radio"/> bar            |
| <input type="radio"/> b                         | <input type="radio"/> foo            |
| <input type="radio"/> c (строка 6)              | <input type="radio"/> main           |
| <input checked="" type="radio"/> c (строка 15)  | <input type="radio"/> -T (строка 15) |
| <input checked="" type="radio"/> c1 (строка 15) | <input type="radio"/> -int (стр. 16) |
| <input type="radio"/> a1                        |                                      |

### 8. Задание

```
class X{
private:
    int* x, y;
    void* z;
public:
    float b, e;
};

int main(){
    X *c, c1;
    int *a1;
}
```

Указателями на объекты являются:

- |                            |                                    |
|----------------------------|------------------------------------|
| <input type="radio"/> x    | <input type="radio"/> x            |
| <input type="radio"/> y    | <input checked="" type="radio"/> c |
| <input type="radio"/> void | <input type="radio"/> c1           |
| <input type="radio"/> z    | <input type="radio"/> a1           |
| <input type="radio"/> b    | <input type="radio"/> int          |
| <input type="radio"/> e    |                                    |

### 9. Задание

Ключевое слово this используется для ...

- ☐ обращения к последней объявленной переменной
- ☐ обращения к базовому классу из производного
- ☐ обращения к производному классу из базового
- ☒ обращения объекта к самому себе
- ☐ для обращения к предку всех классов

### 10. Задание

```
class X{
public:
    float b;
    void bar();
private:
    int* x, y;
    void foo();
};

int main(){
    X o1, *o2;
}
```

Правильные способы обращения к членам класса:

- |         |            |
|---------|------------|
| o1.x    | o1.foo()   |
| o1->x   | o1->foo()  |
| o2.x    | o2.foo()   |
| o2->x   | o2->foo()  |
| ✓ o1.b  | ✓ o1.bar() |
| ✓ o2->b | o2->bar()  |

### 11. Задание

```
class X{
    ...
};
```

Правильные способы объявления одномерного массива из объектов:

- ☐ int X[];
- ☐ int X[128];
- ☐ int[] X;
- ☐ int X[];
- ☐ X \* a;
- ☐ X\* a;
- ☒ X a[128];
- ☒ X \*a[128];
- ☒ X \*a = new X[128];

### 12. Задание

Правильные способы динамического выделения памяти для объекта:

- ☐ SomeClass c = new SomeClass();
- ☒ SomeClass \*c = new SomeClass();
- ☒ SomeClass\* c = new SomeClass();
- ☐ SomeClass \*c = SomeClass();
- ☐ SomeClass\* c = SomeClass();
- ☐ SomeClass c = SomeClass();
- ☐ SomeClass &c = SomeClass();
- ☐ SomeClass& c = SomeClass();
- ☐ SomeClass\* c();
- ☐ SomeClass c();
- ☐ SomeClass& c();

### 13. Задание

Отношения между классами

```
class X{
    float a;
    float bar();
public:
    void foo();
};
```

```
class Y{
    float b;
    X x;
```

```
public:
    void baz();
};
```

- ☐ класс X - базовый, Y - производный
- ☐ класс Y - базовый, X - производный
- ☐ классы X и Y независимы
- ☐ класс X агрегирует класс Y
- ☒ класс Y агрегирует класс X
- ☐ X ассоциируется с Y
- ☐ класс baz ассоциируется с Y, bar и foo ассоциируются с X
- ☐ класс Y ассоциируется с baz, класс X ассоциируется с bar и foo

### 14. Задание

Отношения между классами

```
class X{
    float a;
    float bar();
public:
    void foo();
};
```

```
class Y: public X{
    float b;
public:
    void baz();
};
```

- ☒ класс X - базовый, Y - производный
- ☐ класс Y - базовый, X - производный
- ☐ классы X и Y независимы
- ☐ класс X агрегирует класс Y
- ☐ класс Y агрегирует класс X
- ☐ класс X наследует bar и foo, Y наследует X
- ☐ класс X наследует bar и foo, Y наследует X и baz

### 15. Задание. Инкапсуляция это ...

- ☐ объединение всех данных в одном классе
- ☒ объединение данных и методов работы с ними в одном классе
- ☐ объединение методов в одном классе
- ☐ разделение методов и данных по разным классам
- ☐ построение одного класса на основе другого
- ☐ включение одного класса в состав другого

### 16. Задание. Наследование это ...

- ☐ замена одного класса другим
- ☐ создание нового класса, методы и поля которого перенесены (закодированы заново или скопированы программистом) из старого класса в новый
- ☒ механизм языка программирования, позволяющий включать члены старого класса в новый
- ☐ объединение двух классов
- ☐ включение одного класса в другой в качестве поля

### 17. Задание

Правильные утверждения о наследовании:

- ☐ закрытые члены класса не наследуются
- ☒ закрытые члены класса наследуются
- ☐ наследуются только открытые члены класса
- ☒ закрытые члены базового класса недоступны в производном



## 18. Задание

Конструктор это ...

- специальная функция, которая создаёт объект
- ☒ • специальный метод инициализирующий объект
- специальный метод инициализирующий объект и выделяющий для него память
- специальная функция, которая создаёт и инициализирует объект
- оператор выделяющий памяти для объекта
- средство IDE позволяющее автоматически создавать классы

## 19. Задание

Верные утверждения о конструкторах. Абстрактные классы не рассматриваются.

- ☒ • конструктор без параметров может быть создан компилятором автоматически
  - В конечном итоге (после компилирования) класс может не иметь конструкторов
  - ☒ • В конечном итоге (после компилирования) класс обязан содержать какой-нибудь конструктор
  - Если в классе определён конструктор, то обязательно должен быть описан и деструктор
- создаётся автоматически!*

## 20. Задание. Классы и объекты

Правильные утверждения о классе и объекте ...

- ☒ • Один класс может соответствовать нескольким объектам
- ☒ • Объект – это экземпляр класса
- Класс – это экземпляр объекта
- Один объект может быть экземпляром одновременно нескольких независимых классов
- Классы состоят из объектов
- Объекты состоят из классов
- ☒ • Класс – это тип данных
- Объект – это тип данных

## 21. Задание

Какие бывают отношения между классами?

- ☒ • наследование
- инкапсуляция
- полиморфизм
- абстрагирование
- ☒ • агрегация (композиция)
- ☒ • ассоциация
- конкатенация

## 22. Задание

- При создании классов следует объединять в каждом классе как можно больше данных и методов
- Классы стоит создавать только для хранения данных
- ☒ • Классы создаются для описания сущностей предметной области
- Стоит стремиться к объединению кода отвечающий за интерфейс пользователя и за бизнес-логику
- ☒ • Класс может описывать как абстрактные понятия так и физические предметы
- Классы предпочтительнее объектов
- Объекты предпочтительнее классов

## 23. Задание. Равенство и идентичность.

- ☒ • объекты одного класса равны если их поля соответственно равны
- объекты одного класса равны если содержат одинаковые методы
- ☒ • объекты одного класса идентичны если их поля соответственно равны
- ☒ • объекты одного класса идентичны если указывают на одну и ту же область памяти

## 24. Задание. Полиморфизм

Выберете правильные утверждения о динамическом полиморфизме (ООП)

- Полиморфизм – механизм языка программирования для построения одних классов на основе других
- Полиморфизм – механизм изменения полей и методов класса
- ☒ • Полиморфизм позволяет вызывать методы с одинаковым названием для разных подтипов
- ☒ • Для организации полиморфизма используется динамическая типизация

## 25. Задание. Абстрактные классы

- нельзя создать экземпляр абстрактного класса
- абстрактный класс – это реализация абстрактного типа данных на языке программирования
- ☒ • абстрактный класс не содержит реализаций для своих методов
- абстрактный класс содержит в себе только данные

## 26. Задание. Массивы.

```
class X{
public:
    void foo();
};
X * array = new X[10];
```

Правильные способы вызова метода из массива объектов:

- X.foo();
- X+.foo;
- X.foo();
- X.foo;
- array()
- array+.foo();
- array[5].foo(); *вызывает ошибку*
- array[4].+.foo();

## 27. Указатели на методы

func – тип описывающий указатель на метод класса T.

Выберите правильные описания типа func, если метод может быть определен как void foo(float x):

- using func = (T::\*)foo(float x);
- using func = (T::\*)(float);
- using func = (T::\*)void (float);
- using func = void (T::\*)foo(float);
- using func = void (T::\*)(float);

## 28. Наиболее правильно описан метод

```
class TestResult{
private:
    short score; // баллы за тест
public:
```

- void set\_score(short s) { score = s; }
- void set\_score(short s) { if (s >= 0) score = s; }
- void set\_score(short s) { cin >> s; score = s; }
- void set\_score() { short s; cin >> s; if (s >= 0) score = s; }
- void get\_score() { short s; cin >> s; if (s >= 0) score = s; }
- void get\_score(short s) { cin >> s; score = s; }

## 29. Наиболее правильно описан метод

```
class TestResult{
private:
    short score; // баллы за тест
public:
```

- void get\_score(short s) { if (s>0) short= s; }
- void get\_score(short s) { cin >> s; score = s; }
- short get\_score(short s) { return score; }
- short get\_score(short s) { if (s >= 0 ) return score; }
- short get\_score() { return score; }
- void get\_score() { short s; cin >> s; if (s>0) score = s; }

## 30. Напишите класс для представления результатов теста. Создайте объект. Заполните его своим именем и предполагаемым результатом.

+3 балла за угаданный результат

```
class TestResult{

    short score; // баллы за тест
    string student_name;
```