

## Тестовое задание для диагностического тестирования по дисциплине

### Объектно-ориентированное программирование

Направление подготовки	<hr/> 09.03.04 <i>шифр</i> Программная инженерия <hr/> <i>наименование</i>
Направленность (профиль)	<hr/> Программное обеспечение компьютерных систем <hr/> <i>наименование</i>
Квалификация выпускника	<hr/> бакалавр <hr/> <i>бакалавр, магистр, специалист</i>
Форма обучения	<hr/> заочная <hr/> <i>очная, заочная, очно-заочная</i>
Кафедра- разработчик	<hr/> автоматики и компьютерных систем <hr/> <i>наименование</i>
Выпускающая кафедра	<hr/> автоматики и компьютерных систем <hr/> <i>наименование</i>

№	Проверяемая компетенция	Тип вопроса	Задание	Варианты ответов	Тип сложности вопроса	Количество баллов за правильный ответ
1.	УК-1.2 ОПК-2.1 ОПК-6.1 ОПК-6.3	Множественный выбор	Каких типов данных не существует в языке С++?	<ul style="list-style-type: none"> <li>– unsigned char</li> <li>– long double</li> <li>– long char</li> <li>– short int</li> <li>– short float</li> <li>– unsigned double</li> </ul>	низкий	5
2.	УК-1.2 ОПК-2.1 ОПК-6.1 ОПК-6.3	Вставить слово	Зарезервированное слово в С++ является _____ (1–4) и его использование _____ (5–6) в каком-либо ином смысле, кроме определенного в _____ (7–9).	<ul style="list-style-type: none"> <li>1: ключевым словом</li> <li>2: идентификатором</li> <li>3: оператором</li> <li>4: константой</li> <li>5: не допускается</li> <li>6: допускается</li> <li>7: данном языке</li> <li>8: данной программе</li> <li>9: данном классе</li> </ul>	низкий	5
3.	УК-1.2 ОПК-2.1 ОПК-6.1 ОПК-6.3	Один из	Что в С++ означает следующее выражение: <code>cout &lt;&lt; x &lt;&lt; endl;</code>	<ul style="list-style-type: none"> <li>– значение переменной x находится строго между значениями переменных cout и endl</li> <li>– значение переменной x будет выведено на экран с последующим переходом на новую строку</li> <li>– двоичный сдвиг значений переменных cout и x на endl разрядов</li> <li>– такое выражение не имеет смысла в С++</li> </ul>	низкий	5

4.	УК-1.2 ОПК-2.1 ОПК-6.1 ОПК-6.3	Один из	Какое определение термина «лексема» является правильным?	<ul style="list-style-type: none"> <li>– минимальная единица языка, имеющая самостоятельный смысл</li> <li>– законченное действие в программе</li> <li>– новый тип данных, определенный разработчиком в тексте программы</li> <li>– элемент общедоступного интерфейса класса</li> </ul>	низкий	5
5.	УК-1.2 ОПК-2.1 ОПК-6.1 ОПК-6.3	Один из	Какую функцию должна содержать любая программа на языке С++?	<ul style="list-style-type: none"> <li>– main()</li> <li>– main++()</li> <li>– cpp_main()</li> <li>– public static void main()</li> </ul>	низкий	5
6.	УК-1.2 ОПК-2.1 ОПК-6.1 ОПК-6.3	Все или ничего	Каких модификаторов доступа не существует в С++?	<ul style="list-style-type: none"> <li>– protected</li> <li>– public</li> <li>– included</li> <li>– private</li> <li>– priority</li> </ul>	средний	5
7.	УК-1.2 ОПК-2.1 ОПК-6.1 ОПК-6.3	Вставить слово	<p>Вставьте пропущенные строки в следующей программе на С++:</p> <pre> _____ (1–3) _____ (4–6) int main() {     cout &lt;&lt; "Hello, World!" &lt;&lt; endl;     _____ (7–9) } </pre>	<pre> 1: #include &lt;stdio.h&gt; 2: #include &lt;cstdio&gt; 3: #include &lt;iostream&gt; 4: using std library; 5: using namespace std; 6: import std.* 7: return 0; 8: end main; 9: break; </pre>	средний	5

8.	УК-1.2 ОПК-2.1 ОПК-6.1 ОПК-6.3	Один из	Что такое конструктор в C++?	<ul style="list-style-type: none"> <li>– специальный метод класса, который выделяет память при создании объекта</li> <li>– специальный метод класса, который предназначен для инициализации полей данных объекта некоторыми начальными значениями</li> <li>– набор элементов, из которых конструируются объекты классов</li> <li>– способ описания новых типов данных</li> </ul>	средний	5
9.	УК-1.2 ОПК-2.1 ОПК-6.1 ОПК-6.3	Один из	В программе массив был создан динамически следующим образом: <code>int *p = new int[1000];</code> Как правильно его уничтожить?	<ul style="list-style-type: none"> <li>– <code>delete p;</code></li> <li>– <code>delete[] p;</code></li> <li>– <code>delete p[];</code></li> <li>– <code>delete p[1000]</code></li> <li>– <code>delete int[1000]</code></li> <li>– <code>free(p);</code></li> <li>– он уничтожится автоматически после использования</li> <li>– такой массив нельзя уничтожить</li> </ul>	средний	5
10.	УК-1.2 ОПК-2.1 ОПК-6.1 ОПК-6.3	Один из	Что такое инкапсуляция?	<ul style="list-style-type: none"> <li>– Свойство языка программирования, позволяющее объединить данные и методы в одном объекте и защитить их от внешнего вмешательства</li> <li>– Возможность использовать одно и то же имя для нескольких объектов или функций</li> <li>– Обобщения множества классов и выделение общего базового класса</li> <li>– Защита текста программы от наблюдения и взлома</li> </ul>	средний	5

11.	УК-1.2 ОПК-2.1 ОПК-6.1 ОПК-6.3	Вставить слово	Как правильно вызвать метод Out() объекта класса Something в каждом из случаев: <pre> class Something {     ...     public:     void Out() const;     void Do     {         _____     } }; void func(const Something &amp; a) {     _____ } int main() {     Something *a = new Something;     _____     return 0; } </pre>	1: a.Out(); 2: a->Out(); 3: Out(); 4: a::Out(); 5: здесь метод Out() недоступен	средний	5
12.	УК-1.2 ОПК-2.1 ОПК-6.1 ОПК-6.3	Один из	Сколько параметров можно передать в деструктор?	– любое количество – ровно один параметр – 0 или 1 параметр в зависимости от ситуации – ни одного, деструктор не имеет параметров	средний	5

13.	УК-1.2 ОПК-2.1 ОПК-6.1 ОПК-6.3	Один из	Что такое полиморфизм?	<ul style="list-style-type: none"> <li>– Механизм, который объединяет данные и код в одном объекте и защищающий их от внешнего вмешательства</li> <li>– Механизм, позволяющий единообразно обрабатывать объекты разных типов за счет наличия у них одного и того же интерфейса</li> <li>– Механизм, позволяющий использовать одно и то же имя для нескольких объектов или функций</li> <li>– Механизм, который позволяет обращаться к одному объекту, используя разные имена</li> </ul>	средний	5
14.	УК-1.2 ОПК-2.1 ОПК-6.1 ОПК-6.3	Один из	Перегрузкой функций называют:	<ul style="list-style-type: none"> <li>– вызов функции с передачей ей большего количества параметров, чем она может принять</li> <li>– использование одного имени для нескольких функций, отличающихся списком формальных параметров</li> <li>– слишком частые вызовы одной функции, из-за чего она начинает давать сбои</li> <li>– реализацию в одной функции слишком сложного алгоритма, который требует много времени для его выполнения</li> </ul>	средний	5

15.	УК-1.2 ОПК-2.1 ОПК-6.1 ОПК-6.3	Один из	Что такое виртуальный метод?	<ul style="list-style-type: none"> <li>– Метод, который не имеет тела (отсутствует реализация метода)</li> <li>– Метод базового класса, который может быть переопределен (замещен) в производном классе</li> <li>– Любой метод, который был унаследован от базового класса</li> <li>– Метод, имеющий такое же имя, как сам класс, и поэтому вызываемый неявно</li> </ul>	средний	5
16.	УК-1.2 ОПК-2.1 ОПК-6.1 ОПК-6.3	Все или ничего	Для некоторого класса Number необходимо перегрузить бинарную операцию «+» («плюс», сложение), которая не изменяет значения своих операндов. Какие из приведенных вариантов являются правильными?	<ul style="list-style-type: none"> <li>– как функцию: Number operator+(const Number &amp; b);</li> <li>– как функцию: Number operator+(const Number &amp; b) const;</li> <li>– как функцию: Number operator+(const Number &amp; a, const Number &amp; b);</li> <li>– как функцию: Number operator+(const Number &amp; a, const Number &amp; b) const;</li> <li>– как метод: Number operator+(const Number &amp; b);</li> <li>– как метод: Number operator+(const Number &amp; b) const;</li> <li>– как метод: Number operator+(const Number &amp; a, const Number &amp; b);</li> <li>– как метод: Number operator+(const Number &amp; a, const Number &amp; b) const;</li> </ul>	высокий	5

17.	УК-1.2 ОПК-2.1 ОПК-6.1 ОПК-6.3	Все или ничего	Чему может соответствовать следующий прототип: Number operator- (Number & x);	<ul style="list-style-type: none"> <li>– перегрузка унарной операции «-» в виде функции</li> <li>– перегрузка унарной операции «-» в виде метода класса</li> <li>– перегрузка бинарной операции «-» в виде функции</li> <li>– перегрузка бинарной операции «-» в виде метода класса</li> </ul>	высокий	5
18.	УК-1.2 ОПК-2.1 ОПК-6.1 ОПК-6.3	Один из	<p>В каком (каких) из блоков будет обработано исключение в следующем фрагменте кода:</p> <pre> try {     throw 1; } catch (Error &amp; e) {     ...    // 1 } catch (int e) {     ...    // 2 } catch (...) {     ...    // 3 } </pre>	<ul style="list-style-type: none"> <li>– в блоке 1</li> <li>– в блоке 2</li> <li>– в блоке 3</li> <li>– во всех по очереди</li> <li>– ни в каком из приведенных</li> <li>– здесь не будет исключения</li> </ul>	высокий	5
19.	УК-1.2 ОПК-2.1 ОПК-6.1 ОПК-6.3	Один из	Если в описании класса не указан ни один модификатор доступа, то все его элементы будут являться:	<ul style="list-style-type: none"> <li>– частными (закрытыми)</li> <li>– защищенными</li> <li>– общедоступными (открытыми)</li> <li>– абстрактными</li> <li>– это приведет к ошибке компиляции</li> </ul>	высокий	5

20.	УК-1.2 ОПК-2.1 ОПК-6.1 ОПК-6.3	Один из	В каких случаях на языке C++ правильно объявлен класс <code>Derive</code> , производный от некоторых классов <code>A</code> и <code>B</code> ?	<ul style="list-style-type: none"> <li>– <code>Derive = class(A, B) { ... };</code></li> <li>– <code>class Derive : public A, public B { ... };</code></li> <li>– <code>class Derive extends A, B { ... };</code></li> <li>– в C++ нет множественного наследования, поэтому объявить такой класс невозможно</li> </ul>	высокий	5
-----	---	---------	---	--	---------	---