Тестовое задание для диагностического тестирования по дисциплине

Объектно-ориентированное программирование

Семестр 3

Код, направление подготовки	09.03.04 Программная инженерия
Направленность (профиль)	Программное обеспечение компьютерных систем
Форма обучения	очная
Кафедра-разработчик	автоматики и компьютерных систем
Выпускающая кафедра	автоматики и компьютерных систем

№	Проверяемая компетенция	Тип вопроса	Задание	Варианты ответов	Тип сложности вопроса
1.	ОПК-2.1 ОПК-6.1 ОПК-6.3	Множественный выбор	Каких типов данных не существует в языке C++?	1. unsigned char 2. long double 3. long char 4. short int 5. short float 6. unsigned double	низкий
2.	ОПК-2.1 ОПК-6.1 ОПК-6.3	Вставить слово	Зарезервированное слово в C++ является _____(1–4) и его использование _____(5–6) в каком-либо ином смысле, кроме определенного в _____(7–9).	1. ключевым словом 2. идентификатором 3. оператором 4. константой 5. не допускается 6. допускается 7. данном языке 8. данной программе 9. данном классе	низкий
3.	ОПК-2.1 ОПК-6.1 ОПК-6.3	Один из	Что в C++ означает следующее выражение: cout << x << endl;	1. значение переменной x находится строго между значениями переменных cout и endl 2. значение переменной x будет выведено на экран с последующим переходом на новую строку 3. двоичный сдвиг значений переменных cout и x на endl разрядов 4. такое выражение не имеет смысла в C++	низкий
4.	ОПК-2.1 ОПК-6.1 ОПК-6.3	Один из	Какое определение термина «лексема» является правильным?	1. минимальная единица языка, имеющая самостоятельный смысл 2. законченное действие в программе 3. новый тип данных, определенный разработчиком в тексте программы 4. элемент общедоступного интерфейса класса	низкий

5.	ОПК-2.1 ОПК-6.1 ОПК-6.3	Один из	Какую функцию должна содержать любая программа на языке C++?	1. main() 2. main++() 3. cpp_main() 4. public static void main()	низкий
6.	ОПК-2.1 ОПК-6.1 ОПК-6.3	Все или ничего	Каких модификаторов доступа не существует в C++?	1. protected 2. public 3. included 4. private 5. priority	средний

7.	ОПК-2.1
	ОПК-6.1
	ОПК-6.3
		Вставить слово	Вставьте пропущенные строки в следующей программе на C++:
			_____(1–3)
			_____(4–6)
			int main()
			{
			cout << "Hello, World!" << endl;
			_____(7–9)
			}
				1. #include <stdio.h>
				2. #include <cstdio>
				3. #include <iostream>
				4. using std library;
				5. using namespace std;
				6. import std.*
				7. return 0;
				8. end main;
				9. break;
					средний

8.	ОПК-2.1 ОПК-6.1 ОПК-6.3	Один из	Что такое конструктор в C++?	1. специальный метод класса, который выделяет память при создании объекта 2. специальный метод класса, который предназначен для инициализации полей данных объекта некоторыми начальными значениями 3. набор элементов, из которых конструируются объекты классов 4. способ описания новых типов данных	средний

9.	ОПК-2.1 ОПК-6.1 ОПК-6.3	Один из	В программе массив был создан динамически следующим образом: int *p = new int[1000]; Как правильно его уничтожить?	1. delete p; 2. delete[] p; 3. delete p[]; 4. delete p[1000] 5. delete int[1000] 6. free(p); 7. он уничтожится автоматически после использования 8. такой массив нельзя уничтожить	средний
10.	ОПК-2.1 ОПК-6.1 ОПК-6.3	Один из	Что такое инкапсуляция?	1. Свойство языка программирования, позволяющее объединить данные и методы в одном объекте и защитить их от внешнего вмешательства 2. Возможность использовать одно и то же имя для нескольких объектов или функций 3. Обобщения множества классов и выделение общего базового класса 4. Защита текста программы от наблюдения и взлома	средний

11.	ОПК-2.1
	ОПК-6.1
	ОПК-6.3
		Вставить слово	Как правильно вызвать метод Out() объекта класса Something в каждом из случаев:
			class Something
			{
			…
			public:
			void Out() const;
			void Do
			{
			_____
			}
			};
			void func(const Something & a)
			{
			_____
			}
			int main()
			{
			Something *a = new Something;
			_____
			return 0;
			}
				1. a.Out();
				2. a->Out();
				3. Out();
				4. a::Out();
				5. здесь метод Out() недоступен
					средний

12.	ОПК-2.1 ОПК-6.1 ОПК-6.3	Один из	Сколько параметров можно передать в деструктор?	1. любое количество 2. ровно один параметр 3. 0 или 1 параметр в зависимости от ситуации 4. ни одного, деструктор не имеет параметров	средний

13.	ОПК-2.1 ОПК-6.1 ОПК-6.3	Один из	Что такое полиморфизм?	1. Механизм, который объединяет данные и код в одном объекте и защищающий их от внешнего вмешательства 2. Механизм, позволяющий единообразно обрабатывать объекты разных типов за счет наличия у них одного и того же интерфейса 3. Механизм, позволяющий использовать одно и то же имя для нескольких объектов или функций 4. Механизм, который позволяет обращаться к одному объекту, используя разные имена	средний
14.	ОПК-2.1 ОПК-6.1 ОПК-6.3	Один из	Перегрузкой функций называют:	1. вызов функции с передачей ей большего количества параметров, чем она может принять 2. использование одного имени для нескольких функций, отличающихся списком формальных параметров 3. слишком частые вызовы одной функции, из-за чего она начинает давать сбои 4. реализацию в одной функции слишком сложного алгоритма, который требует много времени для его выполнения	средний
15.	ОПК-2.1 ОПК-6.1 ОПК-6.3	Один из	Что такое виртуальный метод?	1. Метод, который не имеет тела (отсутствует реализация метода) 2. Метод базового класса, который может быть переопределен (замещен) в производном классе 3. Любой метод, который был унаследован от базового класса 4. Метод, имеющий такое же имя, как сам класс, и поэтому вызываемый неявно	средний

16.	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?	1. как функцию: Number operator+ (const Number & b); 2. как функцию: Number operator+ (const Number & b) const; 3. как функцию: Number operator+ (const Number & a, const Number & b); 4. как функцию: Number operator+ (const Number & a, const Number & b) const; 5. как метод: Number operator+ (const Number & b); 6. как метод: Number operator+ (const Number & b) const; 7. как метод: Number operator+ (const Number & a, const Number & b); 8. как метод: Number operator+ (const Number & a, const Number & b) const;	высокий
17.	ОПК-2.1 ОПК-6.1 ОПК-6.3	Все или ничего	Чему может соответствовать следующий прототип: Number operator- (Number & x);	1. перегрузка унарной операции «-» в виде функции 2. перегрузка унарной операции «-» в виде метода класса 3. перегрузка бинарной операции «-» в виде функции 4. перегрузка бинарной операции «-» в виде метода класса	высокий

18.	ОПК-2.1
	ОПК-6.1
	ОПК-6.3
		Один из	В каком (каких) из блоков будет обработано исключение в следующем фрагменте кода:
			try
			{
			throw 1;
			}
			catch (Error & e)
			{
			… // 1
			}
			catch (int e)
			{
			… // 2
			}
			catch (…)
			{
			… // 3
			}
				1. в блоке 1
				2. в блоке 2
				3. в блоке 3
				4. во всех по очереди
				5. ни в каком из приведенных
				6. здесь не будет исключения
					высокий

19.	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:	1. частными (закрытыми) 2. защищенными 3. общедоступными (открытыми) 4. абстрактными 5. это приведет к ошибке компиляции	высокий
20.	ОПК-2.1 ОПК-6.1 ОПК-6.3	Один из	В каких случаях на языке C++ правильно объявлен класс Derive, производный от некоторых классов A и B?	1. Derive = class(A, B) { … }; 2. class Derive : public A, public B { … }; 3. class Derive extends A, B { … }; 4. в C++ нет множественного наследования, поэтому объявить такой класс невозможно	высокий