

Тестовое задание для диагностического тестирования по дисциплине

Объектно-ориентированное программирование, 5 семестр

Код, направление подготовки	09.03.01 Информатика и вычислительная техника
Направленность (профиль)	ИИиЭС
Форма обучения	Очная
Кафедра разработчик	Автоматизированных систем обработки информации и управления
Выпускающая кафедра	Автоматизированных систем обработки информации и управления

№	Проверяемая компетенция	Задание	Варианты ответов	Тип сложности вопроса
1	ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-8.1 ОПК-8.2 ОПК-8.3 ПК-3.1 ПК-3.2 ПК-3.3 ПК-6.1 ПК-6.2 ПК-6.3 ПК-7.1 ПК-7.2 ПК-7.3 ПК-11.1 ПК-11.2 ПК-11.3	Какой язык является объектно-ориентированным	1. Prolog 2. Assembler 3. C 4. C++	Низкий

2	ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-8.1 ОПК-8.2 ОПК-8.3 ПК-3.1 ПК-3.2 ПК-3.3 ПК-6.1 ПК-6.2 ПК-6.3 ПК-7.1 ПК-7.2 ПК-7.3 ПК-11.1 ПК-11.2 ПК-11.3	Объект - это ...	—	Низкий
3	ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-8.1 ОПК-8.2 ОПК-8.3 ПК-3.1 ПК-3.2 ПК-3.3 ПК-6.1 ПК-6.2 ПК-6.3 ПК-7.1 ПК-7.2 ПК-7.3 ПК-11.1 ПК-11.2 ПК-11.3	Класс - это ...	—	Низкий

4	<p>ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-8.1 ОПК-8.2 ОПК-8.3 ПК-3.1 ПК-3.2 ПК-3.3 ПК-6.1 ПК-6.2 ПК-6.3 ПК-7.1 ПК-7.2 ПК-7.3 ПК-11.1 ПК-11.2 ПК-11.3</p>	<p>Объектно-ориентированное программирование - это ...</p>	<p>1. методология программирования, основанная на представлении программы в виде совокупности объектов, каждый из которых является экземпляром определённого класса, а классы образуют иерархию наследования 2. методология программирования, основанная на представлении программы в виде модулей 3. методология программирования, основанная на представлении программы в виде совокупности логических функций 4. методология программирования, основанная на представлении программы в виде совокупности моделей, каждый из которых является экземпляром определённого шаблона, а шаблоны образуют иерархию наследования</p>	<p>Низкий</p>
---	--	--	--	---------------

5	ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-8.1 ОПК-8.2 ОПК-8.3 ПК-3.1 ПК-3.2 ПК-3.3 ПК-6.1 ПК-6.2 ПК-6.3 ПК-7.1 ПК-7.2 ПК-7.3 ПК-11.1 ПК-11.2 ПК-11.3	В объектно-ориентированном программировании число является	1. переменной 2. типом 3. полем 4. объектом	Низкий
6	ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-8.1 ОПК-8.2 ОПК-8.3 ПК-3.1 ПК-3.2 ПК-3.3 ПК-6.1 ПК-6.2	Наследование - это ...	1. механизм объектно-ориентированного программирования, согласно которой абстрактный тип данных может наследовать данные и функциональность некоторого существующего типа, способствуя повторному использованию компонентов программного обеспечения 2. концепция объектно-ориентированного программирования, согласно которой абстрактный тип данных может наследовать данные и функциональность некоторого существующего типа, способствуя повторному использованию компонентов программного обеспечения 3. концепция или механизм объектно-ориентированного	Средний

	ПК-6.3 ПК-7.1 ПК-7.2 ПК-7.3 ПК-11.1 ПК-11.2 ПК-11.3		ориентированного программирования, согласно которой абстрактный тип данных может наследовать данные и функциональность некоторого существующего типа, способствуя повторному использованию компонентов программного обеспечения 4. правила объектно- ориентированного программирования, согласно которой абстрактный тип данных может наследовать данные и функциональность некоторого существующего типа, способствуя повторному использованию компонентов программного обеспечения	
7	ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-8.1 ОПК-8.2 ОПК-8.3 ПК-3.1 ПК-3.2 ПК-3.3 ПК-6.1 ПК-6.2 ПК-6.3 ПК-7.1 ПК-7.2 ПК-7.3 ПК-11.1 ПК-11.2 ПК-11.3	Полиморфизм - это ...	1. способность функции или предиката обрабатывать данные разных типов 2. способность функции обрабатывать данные разных типов 3. способность функции обрабатывать данные разных подтипов 4. способность предиката обрабатывать данные разных типов	Средний

8	ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-8.1 ОПК-8.2 ОПК-8.3 ПК-3.1 ПК-3.2 ПК-3.3 ПК-6.1 ПК-6.2 ПК-6.3 ПК-7.1 ПК-7.2 ПК-7.3 ПК-11.1 ПК-11.2 ПК-11.3	Расшифруйте аббревиатуру SOLID	1. single responsibility, open–closed, Liskov substitution, interface segregation и dependency inversion 2. single responsibility, open–closed, Liskov substitution, interface segregation и dependency injection 3. single data, open–closed, Liskov substitution, interface segregation и dependency inversion 4. single responsibility, open–connect, Liskov substitution, interface segregation и dependency inversion	Средний
9	ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-8.1 ОПК-8.2 ОПК-8.3 ПК-3.1 ПК-3.2 ПК-3.3 ПК-6.1 ПК-6.2 ПК-6.3 ПК-7.1 ПК-7.2 ПК-7.3 ПК-11.1 ПК-11.2 ПК-11.3	Максимальное количество деструкторов в классе	—	Средний

10	ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-8.1 ОПК-8.2 ОПК-8.3 ПК-3.1 ПК-3.2 ПК-3.3 ПК-6.1 ПК-6.2 ПК-6.3 ПК-7.1 ПК-7.2 ПК-7.3 ПК-11.1 ПК-11.2 ПК-11.3	Основные термины объектно-ориентированного программирования	1. граф 2. объект 3. сеть 4. класс	Средний
11	ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-8.1 ОПК-8.2 ОПК-8.3 ПК-3.1 ПК-3.2 ПК-3.3 ПК-6.1 ПК-6.2 ПК-6.3 ПК-7.1 ПК-7.2 ПК-7.3 ПК-11.1 ПК-11.2 ПК-11.3	... - это концепция объектно-ориентированного программирования, согласно которой абстрактный тип данных может наследовать данные и функциональность некоторого существующего типа, способствуя повторному использованию компонентов программного обеспечения	—	Средний

12	ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-8.1 ОПК-8.2 ОПК-8.3 ПК-3.1 ПК-3.2 ПК-3.3 ПК-6.1 ПК-6.2 ПК-6.3 ПК-7.1 ПК-7.2 ПК-7.3 ПК-11.1 ПК-11.2 ПК-11.3	Абстракция это ...	1. использование только эффективных характеристик объекта, которые имеются в данной системе 2. использование только не эффективных характеристик объекта, которые имеются в данной системе 3. использование только тех характеристик объекта, которые с достаточной точностью представляют его в данной системе 4. использование всех характеристик объекта, которые представлены в данной системе	Средний
13	ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-8.1 ОПК-8.2 ОПК-8.3 ПК-3.1 ПК-3.2 ПК-3.3 ПК-6.1 ПК-6.2 ПК-6.3 ПК-7.1 ПК-7.2 ПК-7.3 ПК-11.1 ПК-11.2 ПК-11.3	... - это использование только тех характеристик объекта, которые с достаточной точностью представляют его в данной системе	—	Средний

14	ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-8.1 ОПК-8.2 ОПК-8.3 ПК-3.1 ПК-3.2 ПК-3.3 ПК-6.1 ПК-6.2 ПК-6.3 ПК-7.1 ПК-7.2 ПК-7.3 ПК-11.1 ПК-11.2 ПК-11.3	Инкапсуляция - это ...	1. механизм переадресации, позволяющий осуществлять доступ к различным компонентам 2. правила или утверждения, позволяющий разграничивать доступ к различным частям компонента 3. правила сокрытия, позволяющий разграничивать доступ к различным частям компонента 4. механизм сокрытия, позволяющий разграничивать доступ к различным частям компонента	Средний
15	ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-8.1 ОПК-8.2 ОПК-8.3 ПК-3.1 ПК-3.2 ПК-3.3 ПК-6.1 ПК-6.2 ПК-6.3 ПК-7.1 ПК-7.2 ПК-7.3 ПК-11.1 ПК-11.2 ПК-11.3	Сопоставьте ключевые слова в С#	1. class operator 2. method abstract 3. static virtual	Средний

16	ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-8.1 ОПК-8.2 ОПК-8.3 ПК-3.1 ПК-3.2 ПК-3.3 ПК-6.1 ПК-6.2 ПК-6.3 ПК-7.1 ПК-7.2 ПК-7.3 ПК-11.1 ПК-11.2 ПК-11.3	Расставьте фрагменты кода на С# в правильном порядке	1. } 2. private 3. interface 4. string Name {get;set;} 5. { 6. IAnimal	Высокий
17	ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-8.1 ОПК-8.2 ОПК-8.3 ПК-3.1 ПК-3.2 ПК-3.3 ПК-6.1 ПК-6.2 ПК-6.3 ПК-7.1 ПК-7.2 ПК-7.3 ПК-11.1 ПК-11.2 ПК-11.3	Расставьте фрагменты кода на С# в правильном порядке	1. Car 2. { 3. } 4. } 5. public 6. { 7. class 8. public Car()	Высокий

18	ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-8.1 ОПК-8.2 ОПК-8.3 ПК-3.1 ПК-3.2 ПК-3.3 ПК-6.1 ПК-6.2 ПК-6.3 ПК-7.1 ПК-7.2 ПК-7.3 ПК-11.1 ПК-11.2 ПК-11.3	Расставьте фрагменты кода на С# в правильном порядке	1. Run() 2. } 3. { 4. void 5. public 6. static	Высокий
19	ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-8.1 ОПК-8.2 ОПК-8.3 ПК-3.1 ПК-3.2 ПК-3.3 ПК-6.1 ПК-6.2 ПК-6.3 ПК-7.1 ПК-7.2 ПК-7.3 ПК-11.1 ПК-11.2 ПК-11.3	Расставьте фрагменты кода на С# в правильном порядке	1. class 2. } 3. Jump() 4. internal 5. {} 6. private void 7. Cat 8. {	Высокий

20	ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-8.1 ОПК-8.2 ОПК-8.3 ПК-3.1 ПК-3.2 ПК-3.3 ПК-6.1 ПК-6.2 ПК-6.3 ПК-7.1 ПК-7.2 ПК-7.3 ПК-11.1 ПК-11.2 ПК-11.3	К принципам SOLID относиться	1. single responsibility, open–closed 2. Liskov substitution, interface segregation и dependency inversion 3. Liskov substitution, interface segregation и dependency injection 4. single data, open–closed	Высокий
----	---	------------------------------	--	---------

№	ПРАВИЛЬНЫЕ ОТВЕТЫ
1	C++
2	сущность в адресном пространстве вычислительной системы, появляющаяся при создании экземпляра класса
3	универсальный, комплексный тип данных, состоящий из тематически единого набора полей и методов
4	методология программирования, основанная на представлении программы в виде совокупности объектов, каждый из которых является экземпляром определённого класса, а классы образуют иерархию наследования
5	объектом
6	концепция объектно-ориентированного программирования, согласно которой абстрактный тип данных может наследовать данные и функциональность некоторого существующего типа, способствуя повторному использованию компонентов программного обеспечения
7	способность функции обрабатывать данные разных типов
8	single responsibility, open–closed, Liskov substitution, interface segregation и dependency inversion
9	1 шт.

10	класс; объект
11	Наследование
12	использование только тех характеристик объекта, которые с достаточной точностью представляют его в данной системе
13	Абстракция
14	механизм сокрытия, позволяющий разграничивать доступ к различным частям компонента
15	class abstract; method virtual; static operator
16	private; interface; IAnimal; {; string Name {get;set};; }
17	public; class; Car; {; public Car(); {; }; }
18	public; static; void; Run(); {; }
19	internal; class; Cat; {; private void; Jump(); {; }; }
20	single responsibility, open–closed; Liskov substitution, interface segregation и dependency inversion