

Документ подписан при тестовом задании для  
 Информация о владельце:  
 ФИО: Косенок Сергей Михайлович  
 Должность: ректор  
 Дата подписания: 06.06.2024 06:43:51  
 Уникальный программный ключ:  
 e3a68f3eaa1e02674b5414998099d3d6bfdcf836

**Тестовое задание для диагностического тестирования по дисциплине:**

*Объектно-ориентированное программирование, 5 семестр*

Код направления подготовки	01.03.02 Прикладная математика и информатика
Направленность (профиль)	Прикладная математика и информатика
Форма обучения	очная
Кафедра-разработчик	Кафедра прикладной математики
Выпускающая кафедра	Кафедра прикладной математики

Проверяемая компетенция	Задание	Варианты ответов	Тип сложности вопроса	Кол-во баллов за правильный ответ
ПК-3.1, ПК-3.2, ПК-4.1, ПК-4.2, ПК-4.3	1. Класс - это ...		Низкий	2
ПК-3.1, ПК-3.2, ПК-4.1, ПК-4.2, ПК-4.3	2. Объект - это ...		Низкий	2
ПК-3.1, ПК-3.2, ПК-4.1, ПК-4.2, ПК-4.3	3. Какой язык является объектно-ориентированным	1. Assembler 2. Prolog 3. C 4. C++	Низкий	2

<p>ПК-3.1, ПК-3.2, ПК-4.1, ПК-4.2, ПК-4.3</p>	<p>4.           Объектно-ориентированное программирование - это ...</p>	<p>1. Методология программирования, основанная на представлении программы в виде совокупности логических функций 2. Методология программирования, основанная на представлении программы в виде совокупности моделей, каждый из которых является экземпляром определённого шаблона, а шаблоны образуют иерархию наследования 3. Методология программирования, основанная на представлении программы в виде модулей 4. Методология программирования, основанная на представлении программы в виде совокупности объектов, каждый из которых является экземпляром определённого класса, а классы образуют иерархию наследования</p>	<p>Низкий</p>	<p>2</p>
---	---	---	---------------	----------

ПК-3.1, ПК-3.2, ПК-4.1, ПК-4.2, ПК-4.3	5. В объектно-ориентированном программировании число является	1. Объектом 2. Типом 3. Переменной 4. Поле	Низкий	2
ПК-3.1, ПК-3.2, ПК-4.1, ПК-4.2, ПК-4.3	6. Максимальное количество деструкторов в классе		Средний	5
ПК-3.1, ПК-3.2, ПК-4.1, ПК-4.2, ПК-4.3	7. Сопоставьте ключевые слова в C++	1. class <=> virtual 2. method <=> operator 3. static <=> abstract	Средний	5
ПК-3.1, ПК-3.2, ПК-4.1, ПК-4.2, ПК-4.3	8. Основные термины объектно-ориентированного программирования	1. Класс 2. Граф 3. Сеть 4. Объект	Средний	5

ПК-3.1, ПК-3.2, ПК-4.1, ПК-4.2, ПК-4.3	9. ... - 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	5
ПК-3.1, ПК-3.2, ПК-4.1, ПК-4.2, ПК-4.3	10. ... - это использование только тех характеристик объекта, которые с достаточной точностью представляют его в данной системе		Средний	5

<p>ПК-3.1, ПК-3.2, ПК-4.1, ПК-4.2, ПК-4.3</p>	<p>11. Абстракция - это ...</p>	<p>1. Использование всех характеристик объекта, которые представлены в данной системе  2. Использование только эффективных характеристик объекта, которые имеются в данной системе  3. Использование только не эффективных характеристик объекта, которые имеются в данной системе  4. Использование только тех характеристик объекта, которые с достаточной точностью представляют его в данной системе</p>	<p>Средний</p>	<p>5</p>
---	---------------------------------	--	----------------	----------

<p>ПК-3.1, ПК-3.2, ПК-4.1, ПК-4.2, ПК-4.3</p>	<p>12. Инкапсуляция - это ...</p>	<p>1. Механизм сокрытия, позволяющий разграничивать доступ к различным частям компонента 2. Механизм переадресации, позволяющий осуществлять доступ к различным компонентам 3. Правила или утверждения, позволяющий разграничивать доступ к различным частям компонента 4. Правила сокрытия, позволяющий разграничивать доступ к различным частям компонента</p>	<p>Средний</p>	<p>5</p>
---	-----------------------------------	--	----------------	----------

<p>ПК-3.1, ПК-3.2, ПК-4.1, ПК-4.2, ПК-4.3</p>	<p>13. Наследование - это ...</p>	<p>1. Правила объектно-ориентированного программирования, согласно которой абстрактный тип данных может наследовать данные и функциональность некоторого существующего типа, способствуя повторному использованию компонентов программного обеспечения</p> <p>2. Механизм объектно-ориентированного программирования, согласно которой абстрактный тип данных может наследовать данные и функциональность некоторого существующего типа, способствуя повторному использованию компонентов программного обеспечения</p> <p>3. Концепция объектно-ориентированного программирования, согласно которой абстрактный тип данных может наследовать данные и функциональность некоторого существующего типа, способствуя повторному использованию</p>	<p>Средний</p>	<p>5</p>
---	-----------------------------------	--	----------------	----------

		<p>компонентов программного обеспечения 4. Концепция или механизм объектно- ориентированного программирования, согласно которой абстрактный тип данных может наследовать данные и функциональность некоторого существующего типа, способствуя повторному использованию компонентов программного обеспечения</p>		
--	--	---	--	--



<p>ПК-3.1, ПК-3.2, ПК-4.1, ПК-4.2, ПК-4.3</p>	<p>14. Полиморфизм - это ...</p>	<p>1. Способность функции обрабатывать данные разных типов 2. Способность функции или предиката обрабатывать данные разных типов 3. Способность функции обрабатывать данные разных подтипов 4. Способность предиката обрабатывать данные разных типов</p>	<p>Средний</p>	<p>5</p>
---	--------------------------------------	---	----------------	----------

<p>ПК-3.1, ПК-3.2, ПК-4.1, ПК-4.2, ПК-4.3</p>	<p>15. Расшифруйте аббревиатуру SOLID</p>	<p>1. single data, open–closed, Liskov substitution, interface segregation и dependency inversion 2. single responsibility, open–closed, Liskov substitution, interface segregation и dependency inversion 3. single responsibility, open–closed, Liskov substitution, interface segregation и dependency injection 4. single responsibility, open–connect, Liskov substitution, interface segregation и dependency inversion</p>	<p>Средний</p>	<p>5</p>
---	---	---	----------------	----------

<p>ПК-3.1, ПК-3.2, ПК-4.1, ПК-4.2, ПК-4.3</p>	<p>16. К принципам SOLID относятся</p>	<p>1. Liskov substitution, interface segregation и dependency inversion 2. single responsibility, open–closed 3. Liskov substitution, interface segregation и dependency injection 4. single data, open–closed</p>	<p>Высокий</p>	<p>8</p>
<p>ПК-3.1, ПК-3.2, ПК-4.1, ПК-4.2, ПК-4.3</p>	<p>17. Расставьте фрагменты кода на C++ в правильном порядке</p>	<p>1. Car 2. class 3. public 4. } 5. { 6. public Car() 7. } 8. {</p>	<p>Высокий</p>	<p>8</p>

<p>ПК-3.1, ПК-3.2, ПК-4.1, ПК-4.2, ПК-4.3</p>	<p>18. Расставьте фрагменты кода на C++ в правильном порядке</p>	<p>1. public 2. { 3. void 4. Run() 5. static 6. }</p>	<p>Высокий</p>	<p>8</p>
<p>ПК-3.1, ПК-3.2, ПК-4.1, ПК-4.2, ПК-4.3</p>	<p>19. Расставьте фрагменты кода на C++ в правильном порядке</p>	<p>1. { 2. Cat 3. } 4. {} 5. private void 6. internal 7. Jump() 8. class</p>	<p>Высокий</p>	<p>8</p>

ПК-3.1, ПК-3.2, ПК-4.1, ПК-4.2, ПК-4.3	20. Расставьте фрагменты кода на C++ в правильном порядке	1. } 2. string Name{get;set;} 3. { 4. private 5. IAnimal 6. interface	Высокий	8
--	--	---	---------	---