

Документ подписан простой электронной подписью
Информация о владельце:

ФИО: Косенок Сергей Михайлович

Должность: ректор

Дата подписания: 11.06.2026 09:40:20

Уникальный программный ключ:

e3a68f3eaa1e62674b54f490b09d3461f1f936

Тестовое задание для диагностического тестирования по дисциплине

Объектно-ориентированное программирование, 5, 6 семестр


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

Тестовое задание для диагностического тестирования (5 семестр)

№	Проверяемая компетенция	Задание	Варианты ответов	Тип сложности вопроса
1	ПК-12.2 ПК-12.3 ПК-13.1 ПК-13.2 ПК-13.3	Универсальный, комплексный тип данных, являющийся моделью информационной сущности		Низкий
2	ПК-12.2 ПК-12.3 ПК-13.1 ПК-13.2 ПК-13.3	Сущность в адресном пространстве вычислительной системы, появляющаяся при создании экземпляра класса		Низкий

3	ПК-12.2 ПК-12.3 ПК-13.1 ПК-13.2 ПК-13.3	Объектно-ориентированным является язык	1. Assembler 2. Prolog 3. C 4. C++	Низкий
4	ПК-12.2 ПК-12.3 ПК-13.1 ПК-13.2 ПК-13.3	Объектно-ориентированное программирование – это методология программирования, основанная на представлении программы в виде	1. совокупности логических функций 2. совокупности моделей, каждый из которых является экземпляром определённого шаблона, а шаблоны образуют иерархию наследования 3. модулей 4. совокупности объектов, каждый из которых является экземпляром определённого класса, а классы образуют иерархию наследования	Низкий
5	ПК-12.2 ПК-12.3 ПК-13.1 ПК-13.2 ПК-13.3	В объектно-ориентированном программировании число является	1. объектом 2. типом 3. переменной 4. полем	Низкий
6	ПК-12.2 ПК-12.3 ПК-13.1 ПК-13.2 ПК-13.3	Максимальное количество деструкторов в классе	1. 1 шт. 2. 2 шт. 3. 4 шт. 4. 0 шт.	Средний
7	ПК-12.2 ПК-12.3 ПК-13.1 ПК-13.2 ПК-13.3	Сопоставьте ключевые слова в C#	1. class <=> virtual 2. method <=> operator 3. static <=> abstract	Средний

8	ПК-12.2 ПК-12.3 ПК-13.1 ПК-13.2 ПК-13.3	Основные термины объектно-ориентированного программирования	1. класс 2. граф 3. сеть 4. объект	Средний
9	ПК-12.2 ПК-12.3 ПК-13.1 ПК-13.2 ПК-13.3	... - 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
10	ПК-12.2 ПК-12.3 ПК-13.1 ПК-13.2 ПК-13.3	... - это использование только тех характеристик объекта, которые с достаточной точностью представляют его в данной системе		Средний
11	ПК-12.2 ПК-12.3 ПК-13.1 ПК-13.2 ПК-13.3	Исключительная ситуация возбуждается в случае ___ операции объекта	1. нарушения предусловия 2. рассогласования результатов 3. нарушения постусловия 4. прекращения выполнения	Средний

12	ПК-12.2 ПК-12.3 ПК-13.1 ПК-13.2 ПК-13.3	<p>Инкапсуляция с С# - это ____ доступ к различным частям компонента</p>	<ol style="list-style-type: none"> 1. механизм сокрытия, позволяющий разграничивать 2. механизм переадресации, позволяющий осуществлять 3. правила или утверждения, позволяющие разграничивать 4. правила сокрытия, позволяющие разграничивать 	Средний
13	ПК-12.2 ПК-12.3 ПК-13.1 ПК-13.2 ПК-13.3	<p>На UML диаграмме классов эта стрелка изображает отношение</p> 	<ol style="list-style-type: none"> 1. Реализация 2. Наследование 3. Полиморфизм 4. Агрегация 	Средний
14	ПК-12.2 ПК-12.3 ПК-13.1 ПК-13.2 ПК-13.3	<p>Полиморфизм – это способность ____</p>	<ol style="list-style-type: none"> 1. функции обрабатывать данные разных типов 2. функции или предиката обрабатывать данные разных типов 3. функции обрабатывать данные разных подтипов 4. предиката обрабатывать данные разных типов 	Средний
15	ПК-12.2 ПК-12.3 ПК-13.1 ПК-13.2 ПК-13.3	<p>Аббревиатура SOLID расшифровывается</p>	<ol style="list-style-type: none"> 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 	Средний

16	ПК-12.2 ПК-12.3 ПК-13.1 ПК-13.2 ПК-13.3	Понятия Интерфейс и Реализация имеют отношение к	<ol style="list-style-type: none"> 1. полиморфизму 2. инкапсуляции 3. параллелизму 4. модульности 	Высокий
17	ПК-12.2 ПК-12.3 ПК-13.1 ПК-13.2 ПК-13.3	Расставьте фрагменты кода на С# в правильном порядке	<ol style="list-style-type: none"> 1. Car 2. class 3. public 4. } 5. { 6. public Car() 7. { } 	Высокий
18	ПК-12.2 ПК-12.3 ПК-13.1 ПК-13.2 ПК-13.3	Расставьте фрагменты кода на С# в правильном порядке	<ol style="list-style-type: none"> 1. public 2. { 3. void 4. Run() 5. static 6. } 	Высокий

19	ПК-12.2 ПК-12.3 ПК-13.1 ПК-13.2 ПК-13.3	Расставьте фрагменты кода на С# в правильном порядке	1. { 2. Cat 3. } 4. { } 5. private void 6. internal 7. Jump() 8. class	Высокий
20	ПК-12.2 ПК-12.3 ПК-13.1 ПК-13.2 ПК-13.3	Расставьте фрагменты кода на С# в правильном порядке	1. } 2. string Name{get;set;} 3. { 4. private 5. IAnimal 6. interface	Высокий

Тестовое задание для диагностического тестирования (6 семестр)

№	Проверяемая компетенция	Задание	Варианты ответов	Тип сложности вопроса
1	ПК-12.2 ПК-12.3 ПК-13.1 ПК-13.2 ПК-13.3	Какой язык является объектно-ориентированным	1. Assembler 2. Prolog 3. C++ 4. C	Низкий
2	ПК-12.2 ПК-12.3 ПК-13.1 ПК-13.2 ПК-13.3	Объект - это ...	—	Низкий

3	ПК-12.2 ПК-12.3 ПК-13.1 ПК-13.2 ПК-13.3	Класс - это ...	—	Низкий
---	---	-----------------	---	--------

4	ПК-12.2 ПК-12.3 ПК-13.1 ПК-13.2 ПК-13.3	Объектно-ориентированное программирование - это ...	<p>1. методология программирования, основанная на представлении программы в виде совокупности объектов, каждый из которых является экземпляром определённого класса, а классы образуют иерархию наследования</p> <p>2. методология программирования, основанная на представлении программы в виде совокупности моделей, каждый из которых является экземпляром определённого шаблона, а шаблоны образуют иерархию наследования</p> <p>3. методология программирования, основанная на представлении программы в виде совокупности логических функций</p> <p>4. методология программирования, основанная на представлении программы в виде модулей</p>	Низкий
---	---	---	--	--------

5	ПК-12.2 ПК-12.3 ПК-13.1 ПК-13.2 ПК-13.3	В объектно-ориентированном программировании число является	1. полем 2. объектом 3. типом 4. переменной	Низкий
---	---	--	--	--------

6	ПК-12.2 ПК-12.3 ПК-13.1 ПК-13.2 ПК-13.3	Наследование - это ...	<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 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>4. концепция объектно-ориентированного программирования, согласно которой абстрактный тип данных может наследовать данные и функциональность некоторого существующего</p>	Средний
---	---	---------------------------	---	---------

			типа, способствуя повторному использованию компонентов программного обеспечения	
--	--	--	--	--

7	ПК-12.2 ПК-12.3 ПК-13.1 ПК-13.2 ПК-13.3	Полиморфизм - это ...	<ol style="list-style-type: none"> 1. способность функции или предиката обрабатывать данные разных типов 2. способность функции обрабатывать данные разных подтипов 3. способность предиката обрабатывать данные разных типов 4. способность функции обрабатывать данные разных типов 	Средний
8	ПК-12.2 ПК-12.3 ПК-13.1 ПК-13.2 ПК-13.3	Расшифруйте аббревиатуру SOLID	<ol style="list-style-type: none"> 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 	Средний

9	ПК-12.2 ПК-12.3 ПК-13.1 ПК-13.2 ПК-13.3	Максимальное количество деструкторов в классе	—	Средний
10	ПК-12.2 ПК-12.3 ПК-13.1 ПК-13.2 ПК-13.3	Основные термины объектно-ориентированного программирования	1. класс 2. граф 3. объект 4. сеть	Средний

11	ПК-12.2 ПК-12.3 ПК-13.1 ПК-13.2 ПК-13.3	... - 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	ПК-12.2 ПК-12.3 ПК-13.1 ПК-13.2 ПК-13.3	Абстракция - это ...	<ol style="list-style-type: none"> 1. использование всех характеристик объекта, которые представлены в данной системе 2. использование только тех характеристик объекта, которые с достаточной точностью представляют его в данной системе 3. использование только эффективных характеристик объекта, которые имеются в данной системе 4. использование только не эффективных характеристик объекта, которые имеются в данной системе 	Средний

13	ПК-12.2 ПК-12.3 ПК-13.1 ПК-13.2 ПК-13.3	... - это использование только тех характеристик объекта, которые с достаточной точностью представляют его в данной системе	—	Средний
14	ПК-12.2 ПК-12.3 ПК-13.1 ПК-13.2 ПК-13.3	Инкапсуляция - это ...	<ol style="list-style-type: none"> 1. правила или утверждения, позволяющий разграничивать доступ к различным частям компонента 2. механизм сокрытия, позволяющий разграничивать доступ к различным частям компонента 3. механизм переадресации, позволяющий осуществлять доступ к различным компонентам 4. правила сокрытия, позволяющий разграничивать доступ к различным частям компонента 	Средний

15	ПК-12.2 ПК-12.3 ПК-13.1 ПК-13.2 ПК-13.3	Сопоставьте ключевые слова в С#	1. class ↔ abstract 2. method ↔ operator 3. static ↔ virtual	Средний
16	ПК-12.2 ПК-12.3 ПК-13.1 ПК-13.2 ПК-13.3	Расставьте фрагменты кода на С# в правильном порядке	1. } 2. interface 3. { 4. private 5. IAnimal 6. string Name{get;set;}	Высокий

17	ПК-12.2 ПК-12.3 ПК-13.1 ПК-13.2 ПК-13.3	Расставьте фрагменты кода на С# в правильном порядке	<pre> 1. Car 2. } 3. public Car() 4. { 5. class 6. { 7. public 8. } </pre>	Высокий
18	ПК-12.2 ПК-12.3 ПК-13.1 ПК-13.2 ПК-13.3	Расставьте фрагменты кода на С# в правильном порядке	<pre> 1. { 2. static 3. } 4. public 5. void 6. Run() </pre>	Высокий

19	ПК-12.2 ПК-12.3 ПК-13.1 ПК-13.2 ПК-13.3	Расставьте фрагменты кода на С# в правильном порядке	<ol style="list-style-type: none"> 1. {} 2. private void 3. Cat 4. } 5. internal 6. Jump() 7. class 8. { 	Высокий
20	ПК-12.2 ПК-12.3 ПК-13.1 ПК-13.2 ПК-13.3	К принципам SOLID относиться	<ol style="list-style-type: none"> 1. single data, open–closed 2. Liskov substitution, interface segregation и dependency inversion 3. single responsibility, open–closed 4. Liskov substitution, interface segregation и dependency injection 	Высокий