

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Косенок Сергей Михайлович
Должность: ректор
Дата подписания: 19.06.2024 07:20:13
Уникальный программный ключ:
e3a68f3eaa1e62674b54f4998099d3d6b0dct856

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 | 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. Assembler 2. C++ 3. C 4. Prolog | Низкий |

| | | | | |
|---|---|------------------|---|--------|
| 2 | 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 | 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 | 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 | Объектно-ориентированное программирование - это ... | 1. методология программирования, основанная на представлении программы в виде совокупности моделей, каждый из которых является экземпляром определённого шаблона, а шаблоны образуют иерархию наследования 2. методология программирования, основанная на представлении программы в виде совокупности логических функций 3. методология программирования, основанная на представлении программы в виде совокупности объектов, каждый из которых является экземпляром определённого класса, а классы образуют иерархию наследования 4. методология программирования, основанная на представлении программы в виде модулей | Низкий |
|---|---|---|---|--------|

| | | | | |
|---|---|--|--|--------|
| 5 | 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. объектом 2. переменной 3. типом 4. полем | Низкий |
|---|---|--|--|--------|

| | | | | |
|---|---|------------------------|--|---------|
| 6 | 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 | Наследование - это ... | 1. 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 2. механизм объектно-ориентированного программирования, согласно которой абстрактный тип данных может наследовать данные и функциональность некоторого существующего типа, способствуя повторному использованию компонентов программного обеспечения 3. правила объектно-ориентированного программирования, согласно которой абстрактный тип данных может наследовать данные и функциональность некоторого существующего типа, способствуя повторному использованию компонентов программного обеспечения 4. концепция объектно-ориентированного программирования, согласно которой абстрактный тип данных может наследовать данные и функциональность некоторого существующего | Средний |
|---|---|------------------------|--|---------|

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

| | | | | |
|---|---|--------------------------------|---|---------|
| 7 | 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 ... | <ol style="list-style-type: none"> 1. способность функции обрабатывать данные разных типов 2. способность функции или предиката обрабатывать данные разных типов 3. способность предиката обрабатывать данные разных типов 4. способность функции обрабатывать данные разных подтипов | Средний |
| 8 | 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 | <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–connect, Liskov substitution, interface segregation и dependency inversion 4. single responsibility, open–closed, Liskov substitution, interface segregation и dependency injection | Средний |

| | | | | |
|----|---|---|---|---------|
| 9 | 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 | 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. класс 2. сеть 3. граф 4. объект | Средний |

| | | | | |
|----|---|--|--|---------|
| 11 | 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- ориентированно го программирова ния, согласно которой абстрактный тип данных может наследовать данные и функционально сть некоторого существующего типа, способствуя повторному использованию компонентов программного обеспечения | — | Средний |
| 12 | 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 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 не эффективных характеристик объекта, которые имеются в данной системе 4. использование только эффективных характеристик объекта, которые имеются в данной системе | Средний |

| | | | | |
|----|---|---|---|---------|
| 13 | 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 | 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. механизм сокрытия, позволяющий разграничивать доступ к различным частям компонента 4. правила сокрытия, позволяющий разграничивать доступ к различным частям компонента | Средний |

| | | | | |
|----|---|--|---|---------|
| 15 | 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 | 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. IAnimal 2. string Name {get;set;} 3. } 4. { 5. interface 6. private | Высокий |

| | | | | |
|----|---|--|--|---------|
| 17 | 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. { 3. public Car() 4. } 5. public 6. class 7. { 8. Car | Высокий |
| 18 | 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. } 3. void 4. static 5. public 6. Run() | Высокий |

| | | | | |
|----|---|--|--|---------|
| 19 | 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 | <ol style="list-style-type: none"> 1. } 2. class 3. { 4. private void 5. Cat 6. {} 7. Jump() 8. internal | Высокий |
| 20 | 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 | <ol style="list-style-type: none"> 1. Liskov substitution, interface segregation и dependency injection 2. Liskov substitution, interface segregation и dependency inversion 3. single data, open–closed 4. single responsibility, open–closed | Высокий |