

Документ подписан простой электронной подписью
Информация о владельце:

ФИО: Косенок Сергей Михайлович

Должность: ректор

Дата подписания: 11.06.2026 09:32:23

Уникальный программный ключ:

e3a68f3eaa1e62674b54f4998099d3d6b11ef876

Тестовое задание для диагностического тестирования по дисциплине

Программирование на языках 4 GL, 5, 6 семестр

Код, направление подготовки	09.03.01 Информатика и вычислительная техника
Направленность (профиль)	Искусственный интеллект и экспертные системы
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	Ускорение параллельных вычислений – это:	1. отношение времени последовательного алгоритма ко времени параллельного решения задачи 2. отношение времени последовательного алгоритма ко времени параллельного решения задачи при использовании максимально возможного количества процессоров 3. отношение времени параллельного алгоритма ко времени последовательного решения задачи	Низкий
2	ПК-12.2 ПК-12.3 ПК-13.1 ПК-13.2 ПК-13.3	Дополните, впишите недостающее слово на месте пропусков «Для организации параллельных вычислений _____ в _____»		Низкий

		вычислительных системах с распределенной памятью необходимо выделить информационно _____ фрагменты вычислений, провести их программную реализацию»		
3	ПК-12.2 ПК-12.3 ПК-13.1 ПК-13.2 ПК-13.3	В чем состоят необходимые условия для возможности организации параллельных вычислений:	1. избыточность вычислительных устройств и независимость их функционирования 2. организация режима разделения времени 3. наличие сети передачи данных между процессорами	Низкий
4	ПК-12.2 ПК-12.3 ПК-13.1 ПК-13.2 ПК-13.3	Дополните, впишите недостающее слово на месте пропусков «в режиме _____ параллельных вычислений на _____ обычном последовательном компьютере без использования дополнительных программных средств с визуализацией процесса решения»		Низкий
5	ПК-12.2 ПК-12.3 ПК-13.1 ПК-13.2 ПК-13.3	Модель вычислений – это:	1. ациклический ориентированный граф 2. бинарное дерево 3. циклический ориентированный граф	Низкий
6	ПК-12.2 ПК-12.3 ПК-13.1 ПК-13.2 ПК-13.3	Какие части программы являются последовательными?	1. чтение входных данных с жесткого диска 2. синхронизация в параллельной программе 3. критическая секция в параллельной программе 4. Ничего из выше перечисленного	Средний

7	ПК-12.2 ПК-12.3 ПК-13.1 ПК-13.2 ПК-13.3	Найдите согласно закону Густавсона ускорение масштабирования некоторой параллельной программы, если известно, что время последовательной части программы равно $math$ мс, время части программы, которая может быть распаралелена, равно $math$ мс, количество процессоров равно $math$. Ответ округлите до десятых.		Средний
8	ПК-12.2 ПК-12.3 ПК-13.1 ПК-13.2 ПК-13.3	Чем характеризуется concurrent-программа?	<ol style="list-style-type: none"> 1. программа порождает процессы, выполняющиеся "одновременно" 2. программа порождает не более одного процесса за весь цикл своего выполнения 3. программа может быть исполнена только на нескольких вычислительных машинах, соединенных сетью 4. программа может быть исполнена только на ЭВМ с несколькими физическими процессорами 	Средний
9	ПК-12.2 ПК-12.3 ПК-13.1 ПК-13.2 ПК-13.3	Дополните, впишите недостающие слова на месте пропусков «Под кластером обычно понимается множество отдельных _____, объединенных в сеть, для которых при помощи специальных аппаратно-программных средств обеспечивается возможность		Средний

		унифицированного _____, надежного _____ и _____ эффективного использования		
10	ПК-12.2 ПК-12.3 ПК-13.1 ПК-13.2 ПК-13.3	Какая директива маркирует начало параллельной секции (OpenMP, язык C)?	1. #pragma parallel 2. #pragma init 3. #pragma mp start 4. Ни одна из предложенных	Средний
11	ПК-12.2 ПК-12.3 ПК-13.1 ПК-13.2 ПК-13.3	Дополните, впишите недостающие слова на месте пропусков «Для организации параллельных вычислений в вычислительных системах с распределенной _____ необходимо выделить информационно независимые _____ вычислений, провести их программную реализацию, разместить полученные части программы на разных процессорах и затем организовать _____ взаимодействие между процессорами»		Средний
12	ПК-12.2 ПК-12.3 ПК-13.1 ПК-13.2 ПК-13.3	Рассмотрим задачу перемножения матрицы на вектор. Пусть размер перемножаемой матрицы 100x100. На вычислительной системе все операции сложения и умножения выполняются одинаковое время 2 нсек. Латентности сети 50 нсек. Пропускная способность сети 60 Мбайт/сек. Элементы матрицы имеют тип double и занимают $w = 8$ байт. Если при распараллеливании использовать разделение матрицы на строки чему	1. 5 2. 3,97 3. 2	Средний

		будет равно ускорение при использовании процессоров:	4	
13	ПК-12.2 ПК-12.3 ПК-13.1 ПК-13.2 ПК-13.3	Каким образом в OpenMP можно явно задать число потоков?	1. с помощью переменной OMP_NUM_THREAD 2. с помощью директивы #pragma omp parallel ... num_threads 3. с помощью функции set_num_threads 4. не один из выше перечисленных	Средний
14	ПК-12.2 ПК-12.3 ПК-13.1 ПК-13.2 ПК-13.3	Выберите верное утверждение об атрибуте private(list) директивы omp parallel в OpenMP.	1. атрибут устанавливает, что все нити пользуются одной копией всех переменных и общих блоков, перечисленных в list 2. атрибут устанавливает, что ни одна из нитей, перечисленных в list, не может войти в блок кода ниже данной директивы 3. атрибут устанавливает, что только master thread может иметь доступ к переменным, перечисленным в list 4. Все утверждения не верные	Средний
15	ПК-12.2 ПК-12.3 ПК-13.1 ПК-13.2 ПК-13.3	Рассмотрим задачу перемножения матриц. Пусть размер перемножаемой матрицы 100x100. На вычислительной системе все операции сложения и умножения выполняются одинаковое время 2 нсек. Латентности сети 500 нсек. Пропускная способность сети 50 Мбайт/сек. Элементы матрицы имеют тип double и занимают w = 8 байт. Если	1. 1,8 2. 2,1 3. 3,6	Средний

		при распараллеливании использовать алгоритм Фокса, чему будет равно теоретическое ускорение при использовании 4 процессоров		
16	ПК-12.2 ПК-12.3 ПК-13.1 ПК-13.2 ПК-13.3	Чем характеризуется многопроцессность (concurrency) в контексте параллельных вычислений?	<ol style="list-style-type: none"> 1. обеспечение минимального времени выполнения одной программы 2. первичность пропускной способности 3. не требуется обеспечение максимальной изоляции процессов друг от друга 4. обеспечение как можно более равномерного распределения ресурсов между процессами 	Высокий
17	ПК-12.2 ПК-12.3 ПК-13.1 ПК-13.2 ПК-13.3	Найдите согласно закону Густавсона ускорение масштабирования некоторой параллельной программы, если известно, что время последовательной части программы равно m мс, время части программы, которая может быть распаралелена, равно $math$ мс, количество процессоров равно $math$. Ответ округлите до десятых.		Высокий
18	ПК-12.2 ПК-12.3 ПК-13.1 ПК-13.2 ПК-13.3	В каких ситуациях может быть реализован истинный параллелизм вычислений?	<ol style="list-style-type: none"> 1. вычисления производятся на ЭВМ с одноподъёмным процессором в многозадачной ОС 2. вычисления производятся на ЭВМ с одноподъёмным процессором в однозадачной ОС 	Высокий

			<p>3. вычисления производятся на многопроцессорном устройстве</p> <p>4. для вычислений применяется процессор, поддерживающий физическую векторизацию</p>	
19	<p>ПК-12.2 ПК-12.3 ПК-13.1 ПК-13.2 ПК-13.3</p>	<p>Найдите согласно закону Густавсона ускорение масштабирования некоторой параллельной программы, если известно, что время последовательной части программы равно m мс, время части программы, которая может быть распаралелена, равно nm мс, количество процессоров равно n. Ответ округлите до десятых.</p>		Высокий
20	<p>ПК-12.2 ПК-12.3 ПК-13.1 ПК-13.2 ПК-13.3</p>	<p>Какие факторы препятствуют получению результата с ожидаемой точностью при распараллеливании арифметических расчетов?</p>	<p>1. машинная операция сложения чисел не обладает свойством коммутативности (порядок сложения двух чисел важен)</p> <p>2. распараллеленные алгоритмы реализуются на одноядерном процессоре</p> <p>3. машинная операция умножения чисел не обладает свойством ассоциативности (важен порядок перемножения трёх чисел)</p> <p>4. машинная операция умножения чисел обладает свойством коммутативности (порядок перемножения двух чисел не важен)</p>	Высокий

Тестовое задание для диагностического тестирования (6 семестр)

№	Проверяемая компетенция	Задание	Варианты ответов
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
2	ПК-12.2 ПК-12.3 ПК-13.1 ПК-13.2 ПК-13.3	Объект - это ...	—

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

4	ПК-12.2 ПК-12.3 ПК-13.1 ПК-13.2 ПК-13.3	Объектно-ориентированное программирование - это ...	<ol style="list-style-type: none"> 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 совокупности моделей, каждый из которых является экземпляром определённого шаблона, а шаблоны образуют иерархию наследования 3. методология программирования, основанная на представлении программы в виде совокупности логических функций 4. методология программирования, основанная на представлении программы в виде модулей
---	---	---	---

5	ПК-12.2 ПК-12.3 ПК-13.1 ПК-13.2 ПК-13.3	В объектно-ориентированном программировании число является	<ol style="list-style-type: none">1. полем2. объектом3. типом4. переменной
---	---	--	---

6	ПК-12.2 ПК-12.3 ПК-13.1 ПК-13.2 ПК-13.3	Наследование - это ...	<ol style="list-style-type: none"> 1. правила 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. 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 4. концепция объектно-ориентированного программирования, согласно которой абстрактный тип данных может наследовать данные и функциональность некоторого существующего 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
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

9	ПК-12.2 ПК-12.3 ПК-13.1 ПК-13.2 ПК-13.3	Максимальное количество деструкторов в классе	—
10	ПК-12.2 ПК-12.3 ПК-13.1 ПК-13.2 ПК-13.3	Основные термины объектно-ориентированного программирования	<ol style="list-style-type: none"> 1. класс 2. граф 3. объект 4. сеть

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	—
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

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	—
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

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
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;}

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

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. {
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