

Документ подписан простой электронной подписью  
Информация о владельце:  
ФИО: Прохоров Сергей Григорьевич  
Должность: Председатель УМК  
Дата подписания: 05.09.2024 10:36:36  
Уникальный программный ключ:  
b1cb3ce3b5a8850f04c5b2579bc691893e7a6284

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ**  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования «Казанский национальный исследовательский  
технический университет им. А.Н. Туполева-КАИ»

**Чистопольский филиал «Восток»**  
*(наименование института (факультета, филиала))*

---

**Кафедра компьютерных и телекоммуникационных систем**  
*(наименование кафедры разработчика)*

---

**УТВЕРЖДЕНО:**  
Ученым советом КНИТУ-КАИ  
(в составе ОП ВО)

**КОМПЛЕКТ ОЦЕНОЧНЫХ МАТЕРИАЛОВ**  
по дисциплине (модулю)

**Б1.В.ДВ.05.01 Оптимизация в программно- аппаратных системах**  
*(индекс дисциплины по учебному плану, наименование дисциплины)*

Чистополь 2023

Комплект оценочных материалов по дисциплине (модулю) разработан для обучающихся всех форм обучения по направлению подготовки (специальности):

Код и наименование направления подготовки (специальности)	Направленность (профиль, специализация, магистерская программа)
09.03.01 Информатика и вычислительная техника	Вычислительные машины, комплексы, системы и сети
	Автоматизированные системы обработки информации и управления

Разработчик(и):

Ефимова Юлия Викторовна, доцент, к.п.н.

Комплект оценочных материалов по дисциплине (модулю) рассмотрен на заседании кафедры КиТС, протокол № 8 от 26.05.2023г.

Заведующий кафедрой

Классен Виктор Иванович, д.т.н.

## 1 ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ (МОДУЛЯ)

Текущий контроль успеваемости обеспечивает оценивание хода освоения дисциплины (модуля).

Промежуточная аттестация предназначена для оценки достижения запланированных результатов обучения по завершению изучения дисциплины (модуля) и позволяет оценить уровень и качество ее освоения обучающимися.

Комплект оценочных материалов представляет собой совокупность оценочных средств (комплекс заданий различного типа с ключами правильных ответов, включая критерии оценки), используемых при проведении оценочных процедур (текущего контроля, промежуточной аттестации) с целью оценивания достижения обучающимися результатов обучения по дисциплине (модулю).

### 1.1 Оценочные средства и балльные оценки для контрольных мероприятий

Таблица 1.1 – Объем дисциплины (модуля) для очной формы обучения

Семестр	Общая трудоемкость дисциплины (модуля), в з.е./час	Виды учебной работы, в т.ч. проводимые с использованием ЭО и ДОТ											
		Контактная работа обучающихся с преподавателем по видам учебной работы (аудиторная работа)							Самостоятельная работа обучающегося (внеаудиторная работа)				
		Лекции	Лабораторные работы	Практические занятия	Курсовая работа (консультация, защита)	Курсовой проект (консультация, защита)	Консультации перед экзаменом	Контактная работа на промежуточной аттестации	Курсовая работа (подготовка)	Курсовой проект (подготовка)	Проработка учебного материала (самоподготовка)	Подготовка к промежуточной аттестации	Форма промежуточной аттестации
7	4 ЗЕ/144	16	32		-	-	-	0,35	-	-	60	35,65	экзамен
<b>Итого</b>	<b>4ЗЕ/144</b>	<b>16</b>	<b>32</b>		<b>-</b>	<b>-</b>	<b>-</b>	<b>0,35</b>	<b>-</b>	<b>-</b>	<b>60</b>	<b>35,65</b>	

Текущий контроль успеваемости и промежуточная аттестация по дисциплине (модулю) осуществляется в соответствии с балльно-рейтинговой

системой по 100-балльной шкале. Балльные оценки для контрольных мероприятий представлены в таблице 1.2. Пересчет суммы баллов в традиционную оценку представлен в таблице 1.3.

Таблица 1.2 Балльные оценки для контрольных мероприятий

Наименование контрольного мероприятия	Максимальный балл на первую аттестацию	Максимальный балл за вторую аттестацию	Максимальный балл за третью аттестацию	Всего за семестр
7 семестр				
Тестирование	6	7	7	20
Выполнение лабораторной работы		15	15	30
Итого (максимум за период)	<b>6</b>	<b>22</b>	<b>22</b>	<b>50</b>
Экзамен				<b>50</b>
Итого				<b>100</b>

Таблица 1.3 Шкала оценки на промежуточной аттестации

Выражение в баллах	Словесное выражение при форме промежуточной аттестации - зачет	Словесное выражение при форме промежуточной аттестации – экзамен, зачет с оценкой
от 86 до 100	Зачтено	Отлично
от 71 до 85	Зачтено	Хорошо
от 51 до 70	Зачтено	Удовлетворительно
до 51	Не зачтено	Неудовлетворительно

Форма и организация промежуточной аттестации по итогам освоения дисциплины – экзамен, проводится два этапа: тестирование и устные ответы на экзаменационные вопросы.

## 1.2 Оценочные средства для проведения текущего контроля

### 1.2.1 Тестовые вопросы

Тестовые вопросы содержат следующие типы вопросов с соответствующим количеством баллов за правильный ответ:

Тип вопроса	Количество баллов за правильный ответ
запрос выбора вариантов ответа	1
запрос нескольких ответов	1 -при выборе всех правильных 0,5 – за 2 правильных из 3 0,25 – за 1 правильный из 3 0,5 – за 1 правильный из 2
запрос ввода пропущенного текста	1

Объект ThreadPool позволяет:

- (1) автоматизировать распределение пользовательских задач по рабочим потокам пула+
- (2) получить информацию о распараллеливании выполнения пользовательских задач
- (3) настраивать тип декомпозиции пользовательских задач
- (4) нет правильных вариантов

Блочная (динамическая) декомпозиция позволяет эффективнее обрабатывать:

- (1) итерации с малой вычислительной нагрузкой
- (2) элементы структуры с равной вычислительной нагрузкой обработки каждого элемента
- (3) элементы структуры с неравномерной вычислительной нагрузкой обработки каждого элемента+
- (4) блочная декомпозиция не является эффективной

Декомпозиция по диапазону позволяет эффективнее обрабатывать:

- итерации с малой вычислительной нагрузкой  
итерации с большой вычислительной нагрузкой
- (2) элементы структуры с равной вычислительной нагрузкой обработки каждого элемента+
  - (3) элементы структуры с неравномерной вычислительной нагрузкой обработки каждого элемента
  - (4) декомпозиция по диапазону не является эффективной

Выберете утверждения, соответствующие многоядерной архитектуре

- (1) вычислительная система содержит несколько исполнительных устройств, работающих с общей памятью+
- (2) вычислительная система содержит одно исполнительное устройство и несколько блоков кэшируемой памяти
- (3) вычислительная система содержит несколько исполнительных устройств, каждое из которых работает со своей оперативной памятью
- (4) нет правильных ответов

Параллельный алгоритм состоит из двух подзадач и обеспечивает ускорение в два раза на двудерной системе. Чему равна эффективность алгоритма системе с четырьмя процессорами?

- (1) 0.25
- (2) 0.5+
- (3) 1
- (4) 2

В каких случаях возникает проблема гонки данных:

- (1) потоки осуществляют запись в разделяемую переменную
- (2) один поток осуществляет запись в разделяемую переменную, а другой поток читает разделяемую переменную
- (3) потоки увеличивают общий счетчик, выполняя оператор инкрементирования+
- (4) два потока добавляют элементы в один список типа очереди (Queue)+

При каких условиях поток находится в состоянии "Готовый"?

- (1) создан экземпляр класса Thread.
- (2) объект типа Thread запущен с помощью метода Start+
- (3) поток выполняется на процессоре (ядре процессора)
- (4) выполняющийся поток выгружен по истечении кванта времени+

Какие средства синхронизации можно использовать для организации взаимно-исключительный доступ к разделяемому ресурсу?

- (1) Mutex+
- (2) SemaphoreSlim+
- (3) lock+
- (4) ManualResetEvent

Выберете правильные утверждения

- (1) каждая задача выполняется в одном рабочем потоке
- (2) несколько задач могут выполняться в одном рабочем потоке+
- (3) задача может выполняться в нескольких рабочих потоках
- (4) ни одно из указанных утверждений не верно

Стратегия Inlined Execution применяется планировщиком

- (1) для запуска задач из локальной очереди потока, если текущая задача заблокирована вызовом ожидания+
- (2) для запуска задач из глобальной очереди, если текущая задача заблокирована
- (3) для заимствования задач свободным потоком из очереди занятого потока
- (4) для запуска задачи в независимом потоке

Выберете утверждения, соответствующие многоядерной архитектуре

- (1) вычислительная система содержит несколько исполнительных устройств, работающих с общей памятью+
- (2) вычислительная система содержит одно исполнительное устройство и несколько блоков кэшируемой памяти
- (3) вычислительная система содержит несколько исполнительных устройств, каждое из которых работает со своей оперативной памятью
- (4) нет правильных ответов

Число рабочих потоков в пуле зависит от следующих факторов:

- (1) возможности вычислительной системы (число процессоров/ядер)+
- (2) загруженность вычислительной системы+
- (3) параметры пула потоков+
- (4) число рабочих элементов, добавленных в пул потоков

В каких случаях задача завершается со статусом Faulted

- (1) во вложенных задачах возникли исключения
- (2) задача была отменена
- (3) в дочерних задачах возникли исключения+
- (4) в задаче возникло исключение+

Для выполнения циклической обработки с большим числом итераций и малой вычислительной нагрузкой на каждой итерации эффективнее использовать:

- (1) Parallel.ForEach с пакетной обработкой итераций+
- (2) Parallel.ForEach с блочной декомпозицией

- (3) `Parallel.Invoke`
- (4) последовательный цикл `for`.

Выберете правильные утверждения, соответствующие системам с общей памятью

- (1) для реализации параллельного выполнения программы обязательно применение средств синхронизации
- (2) при выполнении параллельной программы может возникать ситуация гонки данных+
- (3) применение средств синхронизации позволяет решить проблему гонки данных+
- (4) синхронизация необходима только при выполнении программы в режиме разделения времени

В рабочей функции потока используется переменная `SomeVal` типа `int`. Переменная является локальной для каждого потока в следующих случаях:

- (1) переменная объявлена внутри рабочей функции потока+
- (2) переменная класса, используемого в рабочей функции, объявлена с модификатором `private`.
- (3) переменная класса, используемого в рабочей функции, объявлена с атрибутом `ThreadStaticAttribute`+
- (4) переменная объявлена внутри класса, содержащего рабочую функцию потока

Какие средства синхронизации позволяют организовать барьерную синхронизацию (один поток дожидается завершения нескольких потоков):

- (1) `Lock`
- (2) `Monitor`
- (3) `Mutex`
- (4) `SemaphoreSlim`+

Какой метод объекта `ParallelLoopState` позволяет досрочно выйти из цикла, но обработать все итерации с номером меньшим, чем заданный?

- (1) `Abort`
- (2) `Break`+
- (3) `Stop`
- (4) не существует метода для досрочного выхода из параллельного цикла

Какие операторы выполняются последовательно в PLINQ-запросах



- (1) Min, Max
- (2) Aggregate
- (3) Distinct, GroupBy
- (4) нет правильного ответа+

Параметр LongRunning при создании задачи

- (1) настраивает оптимизационные алгоритмы планировщика по анализу эффективности выполнения задач
- (2) планировщик не отслеживает эффективность выполнения задачи в рабочем потоке
- (3) для задачи создается новый поток, не входящий в число рабочих потоков+
- (4) рабочий поток выполнения задачи получает низкий приоритет
- (5) рабочий поток выполнения задачи получает высокий приоритет

Выберете правильные утверждения, соответствующие свойству масштабируемости параллельного приложения

- (1) приложение обеспечивает ускорение, равное предельному, согласно закону Амдала
- (2) приложение может обрабатывать разные объемы данных
- (3) приложение максимально задействует все имеющиеся вычислительные ресурсы+
- (4) нет правильных вариантов

Вызов метода Thread.Sleep(0) приводит к следующему:

- (1) текущий поток снимается с процессора, не дожидаясь завершения выделенного кванта времени+
- (2) поток продолжает выполняться без промедлений
- (3) текущий поток блокируется на бесконечное время
- (4) генерируется исключение

Объект Monitor поддерживает следующие возможности

- (1) попытка входа в критическую секцию в течение заданного интервала времени+
- (2) попытка входа в критическую секцию+
- (3) вход в критическую секцию нескольких потоков
- (4) синхронизация на уровне операционной системы (межпроцессная синхронизация)

Задачи-продолжения позволяют:

- (1) реализовать обработчик отмены задач+
- (2) запускать задачу при определенном статусе завершения предшествующей задачи+
- (3) запускать задачу при определенном статусе завершения предшествующей задачи
- (4) нет правильных вариантов

При осуществлении пакетной обработки с помощью метода `Parallel.ForEach` делегат обработки вызывается

- (1) на этапе инициализации
- (2) на этапе выполнения обработки для каждого элемента
- (3) только один раз для каждого потока на этапе выполнения обработки+
- (4) на этапе финальной редукции

Блок `try` для обработки исключений, возникающий при выполнении PLINQ-запроса, обрамляет:

- (1) определение запроса
- (2) обращение к методам `ToList`, `ToArray`, `ToDictionary`+
- (3) перебор элементов запроса в `for`- или `foreach`-цикле+
- (4) обработка исключений осуществляется внутри запроса

В чем преимущества использования пула потоков для параллельной обработки?

- (1) рабочие потоки пула являются более легковесными объектами
- (2) нет накладных расходов на создание и завершение потоков+
- (3) распределение рабочих элементов по потокам осуществляется автоматически+
- (4) пул потоков предоставляет встроенные механизмы ожидания завершения работы, механизм отмены

Метод `WaitOne` объекта синхронизации `Semaphore` не приводит к блокировке в случае

- (1) метод `WaitOne` всегда приводит к блокировке потоков в ожидании сигнала об освобождении объекта синхронизации
- (2) объект `Semaphore` инициализирован ненулевым начальным состоянием+
- (3) вызову метода `WaitOne` предшествовал вызов `Release` с ненулевым значением аргумента+

(4) объект Semaphore установлен в ненулевое состояние с помощью метода Set

Для отмены задачи необходимо

- (1) вызвать метод Cancel для объекта типа Task
- (2) вызвать метод Cancel для объекта типа CancellationTokenSource+
- (3) вызвать метод Cancel для объекта типа CancellationToken
- (4) вызвать метод Abort для объекта типа Task

При выполнении параллельной обработки с помощью метода Parallel.Invoke

- (1) число потоков может динамически изменяться
- (2) число потоков задается в качестве аргумента метода Parallel.Invoke
- (3) число потоков может динамически изменяться, но не больше числа ядер вычислительной системы
- (4) число потоков зависит от количества итераций и определяется на этапе инициализации обработки+

Свободный рабочий поток

- (1) обращается к глобальной очереди, если она пуста – обращается к локальной очереди занятого рабочего потока+
- (2) просматривает локальные очереди рабочих потоков, если они пусты – обращается к глобальной очереди
- (3) уничтожается планировщиком
- (4) переходит в спящее состояние

Применение многопроцессного выполнения связано со следующими проблемами:

- (1) гонка данных
- (2) взаимодействие процессов с помощью специальных средств+
- (3) сбой одного процесса приводит к сбою всех процессов
- (4) накладные расходы, связанные с переключением контекста разных процессов+

Сигнальные сообщения типа ManualResetEventSlim позволяют:

- (1) реализовать взаимно-исключительный доступ к фрагменту кода
- (2) реализовать доступ заданного числа потоков к фрагменту кода
- (3) реализовать одновременный запуск всех заблокированных потоков+
- (4) реализовать запуск только одного из ожидающих потоков

Статус задачи `WaitingForActivation` соответствует ситуации:

- (1) задача добавлена в очередь пула потоков и готова к выполнению
- (2) задача объявлена, но было вызова метода `Start`
- (3) задача дожидается завершения вложенных задач
- (4) задача дожидается завершения предшествующих задач+

Для осуществления заимствования рабочим потоком `Thread1` задачи из локальной очереди потока `Thread2` необходимо соблюдение следующих условий:

- (1) глобальная очередь содержит элементы
- (2) глобальная очередь не содержит элементов+
- (3) рабочий поток `Thread1` свободен+
- (4) другие рабочие потоки заняты

Выберите правильные утверждения

- (1) конкурентные коллекции позволяют безопасно добавлять и удалять элементы при одновременной работе в нескольких потоках+
- (2) конкурентные коллекции, не содержащие элементов, блокируют текущий поток при выполнении операции извлечения элемента
- (3) применение конкурентных коллекций идентично применению обычных коллекций с блокировками типа `lock`
- (4) нет правильных ответов

Вызов метода `CompleteAdding` для объекта типа `BlockingCollection<T>` приводит к:

- (1) блокировке текущего потока
- (2) генерации исключения при добавлении элементов+
- (3) генерации исключения при извлечении элемента вне зависимости от числа элементов в коллекции
- (4) освобождению потоков, заблокированных операцией извлечения элементов из пустой коллекции

Метод `Parallel.For` позволяет:

- (1) выполнять агрегированные вычисления+
- (2) осуществлять параллельную обработку независимых итераций+
- (3) осуществлять досрочный выход из цикла+
- (4) распараллеливать произвольный `for`-цикл

В каких случаях не возникает проблема гонки данных

- (1) один поток добавляет элементы в коллекцию типа FIFO (очередь), другой поток извлекает элементы из коллекции
- (2) потоки инкрементируют значения, которые хранятся в разделяемом массиве; каждый поток работает с элементом по уникальному индексу+
- (3) потоки осуществляют запись в разделяемую переменную типа int
- (4) один поток осуществляет запись в разделяемую переменную типа double, другой поток читает значения+

В каких состояниях может находиться поток в процессе работы, если он не выполняет операций ввода-вывода, обращений к функциям ядра и не приостанавливается другими потоками?

- (1) "Готовый"+
- (2) "Ожидающий"
- (3) "Приостановленный"
- (4) "Выполняющийся"+

Какой инструмент синхронизации можно использовать для организации потокобезопасного увеличения счетчика целочисленного типа?

- (1) Monitor+
- (2) Interlocked+
- (3) SpinWait
- (4) ConcurrentBag

Выберете правильные утверждения

- (1) создание задачи приводит к созданию нового потока
- (2) запуск задачи с помощью метода StartNew приводит к созданию потока
- (3) запуск задачи с помощью метода Start приводит к созданию потока
- (4) нет правильных утверждений+

Какие методы работают без явного объявления задач:

- (1) Parallel.Invoke;+
- (2) Task.Factory.StartNew
- (3) ThreadPool.QueueUserWorkItem;+
- (4) Parallel.ForEach+

Выберете правильные утверждения, соответствующие применению синхронизации

- (1) применение синхронизации снижает эффективность параллельного приложения+
- (2) применение синхронизации позволяет обеспечить решение проблемы гонки данных+
- (3) синхронизация является обязательной в параллельных приложениях, использующих многопоточность
- (4) применение синхронизации повышает эффективность параллельного приложения

Второй аргумент в вызове метода пула потоков задает: `ThreadPool.QueueUserWorkItem(f, 5);`

- (1) число потоков, которые будут выполнять метод `f`
- (2) время до запуска метода `f`
- (3) число копий метода `f`, которые добавляются в пул потоков
- (4) аргумент метода `f`+

Какие средства синхронизации используют вызовы ядра операционной системы?

- (1) `Mutex`+
- (2) `ManualResetEvent`+
- (3) `AutoResetEvent`+
- (4) `Monitor`

Блочная (динамическая) декомпозиция позволяет эффективнее обрабатывать:

- (1) итерации с малой вычислительной нагрузкой
- (2) элементы структуры с равной вычислительной нагрузкой обработки каждого элемента
- (3) элементы структуры с неравномерной вычислительной нагрузкой обработки каждого элемента+
- (4) блочная декомпозиция не является эффективной

В каких случаях обеспечивается относительно равномерная загрузка потоков

- (1) объем, выполняемой работы каждым потоком, не одинаков
- (2) осуществляется динамическая (блочная) декомпозиция обработки элементов+
- (3) потоки обрабатывают одинаковое число элементов общей структуры данных
- (4) при функциональной декомпозиции выделяется число подзадач равное числу потоков

Какие средства синхронизации позволяют избежать «мертвой блокировки» ожидающих потоков при возникновении сбоя внутри критической секции?

- (1) lock+
- (2) Semaphore
- (3) Mutex
- (4) AutoResetEvent

Обращение к свойству Result задачи, которая еще выполняется

- (1) блокирует поток до завершения задачи+
- (2) генерирует исключение
- (3) возвращает null
- (4) ни один из предложенных вариантов

Какой метод объекта ParallelLoopState позволяет досрочно выйти из цикла на заданной итерации, вне зависимости от того, какие итерации уже обработаны?

- (1) Abort
- (2) Stop+
- (3) Break
- (4) не существует метода для досрочного выхода из параллельного цикла

Операторы агрегирования для объекта ParallelQuery<T>

- (1) выполняется параллельно
- (2) выполняется последовательно
- (3) могут выполняться как последовательно, так и параллельно+
- (4) объект ParallelQuery<T> не позволяет выполнять операторы агрегирования

Если вложенные задачи выполняются в том же потоке, что и родительская задача, то порядок выполнения следующий:

- (1) в том же порядке, в каком помещаются в локальную очередь
- (2) в обратном порядке+
- (3) порядок выполнения вложенных задач в рабочем потоке является неопределенным

Оптимизация относительного расположения данных в памяти позволяет снять проблему:

- (1) гонка данных

- (2) ложное разделение кэша+
- (3) взаимоблокировка
- (4) сбалансированность нагрузки

Для передачи управления другому потоку, конкурирующему за ядро процессора, на котором выполняется данный поток без прекращения его работы, можно использовать:

- (1) `Thread.Yield`+
- (2) `Thread.Sleep(0)`+
- (3) `Thread.Stop`
- (4) нет средств для передачи управления без прекращения работы текущего потока

Класс `Monitor` поддерживает следующие возможности:

- (1) приостановление потока, выполняющего критическую секцию, на заданный интервал времени
- (2) попытка входа в критическую секцию в течение заданного интервала времени+
- (3) получение информации о заблокированных потоках
- (4) получение информации о потоке, выполняющем критическую секцию

Задачи-продолжения запускаются после завершения задачи предшественника

- (1) вне зависимости от исхода задачи
- (2) если не было отмены задачи предшественника
- (3) если не было исключения в предшествующей задаче
- (4) в зависимости от статуса завершения предшествующей задачи и параметров задачи-продолжения+

При осуществлении пакетной обработки с помощью метода `Parallel.ForEach` синхронизация необходима

- (1) на этапе инициализации
- (2) на этапе выполнения обработки
- (3) на этапе финальной редукции
- (4) необходимость применения синхронизации зависит от выполняемой обработки+

Для ожидания завершения обработки пользовательских рабочих элементов пулом потоков можно использовать

- (1) `ThreadPool.Wait`



- (2) ThreadPool.WaitAll
- (3) Thread.Join
- (4) нет правильных вариантов+

Объект Semaphore обладает следующими возможностями:

- (1) освобождение заданного числа ожидающих потоков+
- (2) инициализация объекта с указанием максимального значения внутреннего счетчика+
- (3) инициализация объекта с указанием начального состояния внутреннего счетчика+
- (4) инициализация объекта с указанием минимального значения внутреннего счетчика

Для обработки исключений, возникающих в задаче, блок try обрамляет:

- (1) вызов метода ожидания задачи t.Wait()+
- (2) вызов метода запуска задачи t.Start()
- (3) объявление задачи Task t = new Task(..);
- (4) нет правильных вариантов

При выполнении параллельной обработки с помощью метода Parallel.For

- (1) число потоков может динамически изменяться
- (2) число потоков задается в качестве аргумента метода Parallel.For
- (3) число потоков определяется количеством итераций на этапе инициализации обработки+
- (4) число потоков может динамически изменяться, но не больше числа ядер вычислительной системы

Число рабочих потоков, участвующих при обработке задач, зависит от следующих факторов::

- (1) вычислительные возможности системы (число ядер)+
- (2) параметры пула потоков+
- (3) результаты обработки предыдущих задач+
- (4) нет правильных ответов

Преимущества использования многопоточности для параллельных вычислений по сравнению с применением многопроцессности связаны:

- (1) потоки работают в едином адресном пространстве+

- (2) переключение потоков одного процесса занимает меньше время по сравнению с переключением потоков разных процессов+
- (3) для взаимодействия потоков не требуется применения средств синхронизации
- (4) сбой в одном потоке не приводит к сбою приложения

Какие методы типа Task и объекта типа Task предполагают блокировку текущего потока до завершения задачи

- (1) t.Wait+
- (2) Task.Read
- (3) t.ContinueWith
- (4) Task.Factory.StartNew

В каком случае не осуществляется заимствование задач из локальной очереди загруженного потока?

- (1) есть задачи в глобальной очереди+
- (2) режим WorkStealing выключен с помощью опций планировщика
- (3) все рабочие потоки заняты обработкой+
- (4) задачи в локальной очереди созданы с опцией PreferFairness

Выберите правильные утверждения

- (1) конкурентные коллекции эффективнее обычных коллекций только в случае параллельной работы+
- (2) конкурентные коллекции функционируют как обычные коллекции при однопоточном использовании
- (3) применение конкурентных коллекций идентично применению обычных коллекций с блокировками типа lock
- (4) нет правильных вариантов

При выполнении операции добавления Add для объекта типа BlockedCollection<T>, содержащего максимальное число элементов и без вызова CompleteAdding, происходит:

- (1) блокировка текущего потока+
- (2) метод возвращает null
- (3) генерируется исключение
- (4) нет правильного ответа

Методы Parallel.For, Parallel.Invoke позволяют

- (1) задать число потоков, которые используются для обработки пользовательских задач
- (2) задать максимальное число потоков, которые используются для обработки пользовательских задач+
- (3) задать минимальное число потоков, которые используются для обработки пользовательских задач
- (4) методы не позволяют регулировать число потоков

В каких случаях не возникает проблема гонки данных

- (1) два потока извлекают элементы из общего списка+
- (2) два потока добавляют элементы в коллекцию типа Dictionary+
- (3) несколько потоков устанавливают значение общей булевой переменной в true перед началом работы
- (4) нет правильных вариантов

В какое состояние может перейти поток после завершения операции ввода-вывода, если до операции поток выполнялся?

- (1) "Приостановленный"
- (2) "Ожидающий"
- (3) "Выполняющийся"
- (4) "Готовый"+

Какие средства синхронизации можно использовать для организации параллельного доступа нескольких потоков к ресурсу?

- (1) Semaphore+
- (2) Mutex
- (3) AutoResetEvent
- (4) lock

Выберете правильные утверждения

- (1) метод `Task.Factory.StartNew` создает новый поток для задачи
- (2) метод `Task.Factory.StartNew` запускает задачу в рабочем потоке и дожидается завершения работы
- (3) метод `Start` запускает задачу в текущем потоке
- (4) метод `Start` добавляет задачу в очередь пула потоков+

Реализация блочной декомпозиции при параллельной обработке с помощью метода `Parallel.ForEach` позволяет

- (1) указать число элементов блока+

- (2) указать начальное число элементов блока
- (3) указать относительные размеры блоков для разных потоков
- (4) указать минимальное число элементов в блоке

Технология Inlined execution предназначена для оптимизации выполнения:

- (1) задач в глобальной очереди
- (2) задач в локально очереди потока+
- (3) только дочерних вложенных задач
- (4) долго выполняющихся задач

Выберете правильные утверждения, соответствующие системам с общей памятью

- (1) для взаимодействия параллельных потоков не требуется специальных интерфейсов передачи сообщений+
- (2) для взаимодействия параллельных частей требуется применять средства синхронизации+
- (3) существует возможность параллелизма как на уровне потоков одного процесса, так и на уровне процессов+
- (4) обеспечивается параллелизм только на уровне приложений как разделение процессорного времени

Для выполнения пользовательских задач с помощью пула потоков необходимо:

- (1) создать экземпляр класса ThreadPool
- (2) задать параметры пула потоков
- (3) использовать метод для добавления рабочих элементов QueueUserWorkItem+
- (4) использовать метод для начала работы пула Start

Какой объект синхронизации можно использовать для предотвращения запуска нескольких копий одного приложения?

- (1) Monitor
- (2) Mutex+
- (3) Interlocked
- (4) SemaphoreSlim

При возникновении необработанного исключения в задаче:

- (1) статус задачи Faulted+
- (2) статус задачи Canceled

- (3) статус задачи не определен
- (4) нет правильных ответов

Локальное хранилище потоков может использоваться для хранения

- (1) только статические поля класса, доступного в рабочей функции потока
- (2) только значимые типы (value types)
- (3) только встроенные типы
- (4) нет правильных ответов+

Какому методу или методам соответствует оператор break при распараллеливании циклической обработки с помощью Parallel.For?

- (1) Abort
- (2) Break+
- (3) Stop+
- (4) для параллельных циклов не существует оператора досрочного выхода

Каким образом выделить для задачи независимый поток, не участвующий в обработке других задач пула?

- (1) использовать параметр LongRunning при создании задачи+
- (2) использовать параметр PreferFairness при создании задачи
- (3) использовать параметр ExecuteSynchronously при создании задачи
- (4) ни один из перечисленных вариантов

Метод Thread.Yield игнорируется в случае:

- (1) нет конкуренции за ядро, на котором выполняется текущий поток+
- (2) нет потоков с большим приоритетом, чем приоритет текущего потока
- (3) приложение выполняется на одноядерном процессоре
- (4) нет правильных ответов

Класс Monitor поддерживает следующие возможности:

- (1) приостановление потока, выполняющего критическую секцию, и освобождение одного из заблокированных потоков+
- (2) попытка входа в критическую секцию в течение заданного интервала времени+
- (3) выполнение критической секции заданным числом потоков
- (4) получение информации о числе заблокированных потоков

При осуществлении агрегированных вычислений с помощью метода Parallel.For синхронизация необходима

- (1) на этапе инициализации
- (2) на этапе выполнения обработки
- (3) на этапе финальной редукции+
- (4) синхронизация не требуется

Можно ли выполнить досрочную остановку выполнения PLINQ-запроса?

- (1) методы Stop и Break объекта ParallelLoopState
- (2) с помощью токена отмены+
- (3) не существует возможности досрочной остановки
- (4) с помощью генерации исключения при обработке элементов

Вложенные задачи выполняются (с учетом действия всех стратегий планировщика):

- (1) в том же рабочем потоке, что и родительская задача
- (2) могут выполняться в разных рабочих потоках+
- (3) если являются дочерними, то выполняются в одном рабочем потоке.
- (4) ни одно из представленных утверждений не верно

Выберете правильные утверждения

- (1) родительская задача всегда дожидается завершения вложенных задач
- (2) родительская задача дожидается только вложенных задач, объявленных с параметром AttachedToParent+
- (3) родительская задача дожидается только дочерних вложенных задач+
- (4) нет правильных утверждений

При выполнении параллельной обработки с помощью метода Parallel.ForEach

- (1) число потоков может динамически изменяться
- (2) число потоков задается в качестве аргумента метода Parallel.ForEach
- (3) число потоков определяется количеством итераций на этапе инициализации обработки+
- (4) число потоков может динамически изменяться, но не больше числа ядер вычислительной системы

Применение многопоточности для параллельных вычислений связано со следующими недостатками:

- (1) для взаимодействия потоков требуется применение специальных средств обмена сообщениями

- (2) для согласования поведения потоков необходимо применение средств синхронизации+
- (3) сбой в одном потоке приводит к сбою приложения+
- (4) многопоточное выполнение приложения связано с накладными расходами, превышающими расходы при многопроцессном выполнении

Сигнальные сообщения типа `ManualResetEvent` позволяют:

- (1) реализовать взаимно-исключительный доступ к фрагменту кода
- (2) реализовать доступ заданного числа потоков к фрагменту кода
- (3) реализовать одновременный запуск всех заблокированных потоков+
- (4) реализовать запуск только одного из ожидающих потоков

Метод `Flatten` применяется для

- (1) преобразования типа `AggregateException` в тип `Exception`
- (2) назначения обработчиков конкретных исключений
- (3) получения списка исключений, возникнувших при исполнении задач+
- (4) нет верных утверждений

Конкурентные коллекции типа `ConcurrentStack<T>`, `ConcurrentQueue<T>`, `ConcurrentBag<T>` можно создавать:

- (1) все конкурентные коллекции можно создавать на базе соответствующих коллекций (`ConcurrentStack<T>` только на базе `Stack<T>`)
- (2) на базе объектов любых типов, реализующих интерфейс `IEnumerable`+
- (3) конкурентные коллекции нельзя создавать на базе других объектов
- (4) нет правильных ответов

### 1.2.2 Выполнение лабораторных работ

Перечень лабораторных работ и система оценивания:

Сем естр	Наименование лабораторной работы	Кол-во баллов	Критерии оценивания
7	1. Знакомство с многопоточной обработкой для оптимизации вычислительных процессов	5	Проведены необходимые опыты и измерения; самостоятельно и рационально выбрано необходимое оборудование; все опыты проведены в условиях и режимах, обеспечивающих получение правильных результатов и выводов; соблюдены требования правил безопасности труда; в отчете правильно и аккуратно выполнены все записи, таблицы, рисунки, чертежи, графики, вычисления; правильно выполнен

			анализ погрешностей.
		4	Работа выполнена полностью. Обучающийся владеет теоретическим материалом, отсутствуют ошибки при описании теории, формулирует собственные, самостоятельные, обоснованные, аргументированные суждения, допуская незначительные ошибки на дополнительные вопросы.
		3	Работа выполнена полностью. Обучающийся владеет теоретическим материалом на минимально допустимом уровне, отсутствуют ошибки при описании теории, испытывает затруднения в формулировке собственных обоснованных и аргументированных суждений, допуская незначительные ошибки на дополнительные вопросы.
		2	Работа выполнена полностью. Обучающийся практически не владеет теоретическим материалом, допуская ошибки по сущности рассматриваемых (обсуждаемых) вопросов, испытывает затруднения в формулировке собственных обоснованных и аргументированных суждений, допускает ошибки при ответе на дополнительные вопросы.
		0-1	Работа выполнена полностью. Обучающийся не владеет теоретическим материалом, допуская грубые ошибки, испытывает затруднения в формулировке собственных суждений, неспособен ответить на дополнительные вопросы.
7	2. Поиск простых чисел	5	Проведены необходимые опыты и измерения; самостоятельно и рационально выбрано необходимое оборудование; все опыты проведены в условиях и режимах, обеспечивающих получение правильных результатов и выводов; соблюдены требования правил безопасности труда; в отчете правильно и аккуратно выполнены все записи, таблицы, рисунки, чертежи, графики, вычисления; правильно выполнен анализ погрешностей.
		4	Работа выполнена полностью. Обучающийся владеет теоретическим материалом, отсутствуют ошибки при описании теории, формулирует собственные, самостоятельные, обоснованные, аргументированные суждения, допуская незначительные ошибки на дополнительные вопросы.
		3	Работа выполнена полностью.



			Обучающийся владеет теоретическим материалом на минимально допустимом уровне, отсутствуют ошибки при описании теории, испытывает затруднения в формулировке собственных обоснованных и аргументированных суждений, допуская незначительные ошибки на дополнительные вопросы.
		2	Работа выполнена полностью. Обучающийся практически не владеет теоретическим материалом, допуская ошибки по сути рассматриваемых (обсуждаемых) вопросов, испытывает затруднения в формулировке собственных обоснованных и аргументированных суждений, допускает ошибки при ответе на дополнительные вопросы.
		0-1	Работа выполнена полностью. Обучающийся не владеет теоретическим материалом, допуская грубые ошибки, испытывает затруднения в формулировке собственных суждений, не способен ответить на дополнительные вопросы.
7	3. Синхронизация доступа к одноэлементному буферу для оптимизации вычислительного процесса	5	Проведены необходимые опыты и измерения; самостоятельно и рационально выбрано необходимое оборудование; все опыты проведены в условиях и режимах, обеспечивающих получение правильных результатов и выводов; соблюдены требования правил безопасности труда; в отчете правильно и аккуратно выполнены все записи, таблицы, рисунки, чертежи, графики, вычисления; правильно выполнен анализ погрешностей.
		4	Работа выполнена полностью. Обучающийся владеет теоретическим материалом, отсутствуют ошибки при описании теории, формулирует собственные, самостоятельные, обоснованные, аргументированные суждения, допуская незначительные ошибки на дополнительные вопросы.
		3	Работа выполнена полностью. Обучающийся владеет теоретическим материалом на минимально допустимом уровне, отсутствуют ошибки при описании теории, испытывает затруднения в формулировке собственных обоснованных и аргументированных суждений, допуская незначительные ошибки на дополнительные вопросы.
		2	Работа выполнена полностью. Обучающийся практически не владеет

			теоретическим материалом, допуская ошибки по сущности рассматриваемых (обсуждаемых) вопросов, испытывает затруднения в формулировке собственных обоснованных и аргументированных суждений, допускает ошибки при ответе на дополнительные вопросы.
		0-1	Работа выполнена полностью. Обучающийся не владеет теоретическим материалом, допуская грубые ошибки, испытывает затруднения в формулировке собственных суждений, неспособен ответить на дополнительные вопросы.
7	4. Синхронизация приоритетного доступа к многоэлементному буферу для оптимизации вычислительного процесса	7-6	Проведены необходимые опыты и измерения; самостоятельно и рационально выбрано необходимое оборудование; все опыты проведены в условиях и режимах, обеспечивающих получение правильных результатов и выводов; соблюдены требования правил безопасности труда; в отчете правильно и аккуратно выполнены все записи, таблицы, рисунки, чертежи, графики, вычисления; правильно выполнен анализ погрешностей.
		5-4	Работа выполнена полностью. Обучающийся владеет теоретическим материалом, отсутствуют ошибки при описании теории, формулирует собственные, самостоятельные, обоснованные, аргументированные суждения, допуская незначительные ошибки на дополнительные вопросы.
		3	Работа выполнена полностью. Обучающийся владеет теоретическим материалом на минимально допустимом уровне, отсутствуют ошибки при описании теории, испытывает затруднения в формулировке собственных обоснованных и аргументированных суждений, допуская незначительные ошибки на дополнительные вопросы.
		2	Работа выполнена полностью. Обучающийся практически не владеет теоретическим материалом, допуская ошибки по сущности рассматриваемых (обсуждаемых) вопросов, испытывает затруднения в формулировке собственных обоснованных и аргументированных суждений, допускает ошибки при ответе на дополнительные вопросы.
		0-1	Работа выполнена полностью. Обучающийся не владеет теоретическим материалом, допуская грубые ошибки, испытывает

			затруднения в формулировке собственных суждений, неспособен ответить на дополнительные вопросы.
7	5. Анализ оптимальности выполнения вычислительного процесса с помощью "Визуализатора параллелизма" в Visual Studio	8-7	Проведены необходимые опыты и измерения; самостоятельно и рационально выбрано необходимое оборудование; все опыты проведены в условиях и режимах, обеспечивающих получение правильных результатов и выводов; соблюдены требования правил безопасности труда; в отчете правильно и аккуратно выполнены все записи, таблицы, рисунки, чертежи, графики, вычисления; правильно выполнен анализ погрешностей.
		6-5	Работа выполнена полностью. Обучающийся владеет теоретическим материалом, отсутствуют ошибки при описании теории, формулирует собственные, самостоятельные, обоснованные, аргументированные суждения, допуская незначительные ошибки на дополнительные вопросы.
		5-4	Работа выполнена полностью. Обучающийся владеет теоретическим материалом на минимально допустимом уровне, отсутствуют ошибки при описании теории, испытывает затруднения в формулировке собственных обоснованных и аргументированных суждений, допуская незначительные ошибки на дополнительные вопросы.
		2	Работа выполнена полностью. Обучающийся практически не владеет теоретическим материалом, допуская ошибки по существу рассматриваемых (обсуждаемых) вопросов, испытывает затруднения в формулировке собственных обоснованных и аргументированных суждений, допускает ошибки при ответе на дополнительные вопросы.
		0-1	Работа выполнена полностью. Обучающийся не владеет теоретическим материалом, допуская грубые ошибки, испытывает затруднения в формулировке собственных суждений, неспособен ответить на дополнительные вопросы.

### 1.2.3 Курсовая работа

Не предусмотрено учебным планом.

### 1.3. Оценочные средства для проведения промежуточного контроля (промежуточной аттестации)

Семестр	Вид промежуточной аттестации	Вид контрольного мероприятия	Балльные оценки
7	Экзамен	Тестовые задания Экзаменационные вопросы	0-20 0-30

#### 1.3.1. Тестовые задания

Тестовые задания промежуточной аттестации представляют собой совокупность тестовых вопросов текущего контроля.

#### 1.3.2 Комплексное задание (экзаменационный билет)

Билеты экзамена равноценны по трудности, одинаковы по структуре, параллельны по расположению заданий. Комплексное экзаменационное задание состоит из 2 вопросов теоретического характера. Теоретический вопрос направлен на проверку знаний.

##### 1.3.2.1 Вопросы на зачете/экзамене (экзаменационные вопросы)

№ п/п	Тип вопроса	Вопрос
1	Теоретический	Особенности сценария обработки для обеспечения оптимизации в программно-аппаратных системах.
2		Отличие декомпозиция по данным отличается от декомпозиции по вычислениям
3		Достоинства и недостатки статического и динамического способов назначения задач виртуальным исполнителям.
4		Проблемы разработки параллельных приложений
5		Работа с потоками. Многопоточная обработка данных
6		Средства синхронизации. Конкурентные коллекции
7		Синхронизация доступа к одноэлементному буферу
8		Синхронизация приоритетного доступа к многоэлементному буферу
9		Достоинства и недостатки различных подмоделей программирования в модели общей памяти.
10		Программирование циклов с помощью <code>Parallel.ForEach()</code>
11		Программирование циклов с помощью <code>Parallel.For()</code>
12		Шаблоны параллелизма <code>Parallel</code>
13		Технология PLINQ Параллельный запрос

14		Параллельные шаблоны агрегирования в PLINQ
15		Визуальные средства отладки параллельных и многопоточных приложений в среде Visual Studio
16		Планировщик задач. Понятие задачи. Параллелизм задач
17		Явное и неявное создание и запуск задач . Ожидание завершения задач
18		Типовые модели параллельных приложений
19		Возможности эквивалентного изменения порядка вложенности во вложенных циклах для распараллеливания программ
20		Средства усовершенствования параллельной отладки

### *Критерии оценивания*

Суммарно оцениваются ответы на вопросы. Ответы должны быть развернутыми, полными. Каждый правильный ответ на вопрос оценивается до 15 баллов в зависимости от полноты ответа.

Оценивается полнота раскрытия материала; логичность изложения материала; умение иллюстрировать конкретными примерами; знание формул, терминологии, обозначений; использование профессиональной терминологии; демонстрация усвоенного ранее материала; самостоятельность в изложении материала.

### *Пример балльной системы оценивания:*

Критерии оценивания	Количество баллов
<ul style="list-style-type: none"> <li>– полно раскрыто содержание материала;</li> <li>– материал изложен грамотно, в определенной логической последовательности;</li> <li>– продемонстрировано системное и глубокое знание материала;</li> <li>– точно используется терминология;</li> <li>– показано умение иллюстрировать теоретические положения конкретными примерами, применять их в новой ситуации;</li> <li>– продемонстрировано усвоение ранее изученных сопутствующих вопросов;</li> <li>– ответ дан самостоятельно, без наводящих вопросов;</li> <li>– продемонстрирована способность творчески применять знание теории к решению профессиональных задач;– допущены одна-две неточности при освещении второстепенных вопросов, которые исправляются по замечанию;</li> </ul>	10-15
<ul style="list-style-type: none"> <li>– вопросы излагаются систематизировано и последовательно;</li> <li>– продемонстрировано умение анализировать материал, однако не все выводы носят аргументированный и доказательный характер;</li> <li>– продемонстрировано усвоение основной литературы;</li> <li>– ответ удовлетворяет в основном требованию на максимальную оценку, но при этом имеет один из недостатков: в изложении допущены небольшие пробелы, не исказившие содержание ответа; допущены один-два недочета при освещении основного содержания ответа, исправленные по замечанию преподавателя;</li> </ul>	7-9

– допущены ошибка или более двух недочетов при освещении второстепенных вопросов, которые легко исправляются по замечанию преподавателя;	
– неполно или непоследовательно раскрыто содержание материала, но показано общее понимание вопроса и продемонстрированы умения, достаточные для дальнейшего усвоения материала; – усвоены основные категории по рассматриваемому и дополнительным вопросам; – имелись затруднения или допущены ошибки в определении понятий, использовании терминологии, исправленные после нескольких наводящих ответов; – неполное знание теоретического материала, обучающийся не может применить теорию в новой ситуации; – продемонстрировано усвоение основной литературы;	4-6
– не раскрыто основное содержание учебного материала либо отказ от ответа; – обнаружено незнание или непонимание большей или наиболее важной части учебного материала; – допущены ошибки в определении понятий, при использовании терминологии, некоторые не исправлены после нескольких наводящих вопросов.	1-3
-ответ не получен.	0

*Пример балльной системы оценивания вопросов:*

Задание	Критерии оценивания	Количество баллов
Теоретический вопрос	– полно раскрыто содержание материала; – материал изложен грамотно, в определенной логической последовательности; – продемонстрировано системное и глубокое знание материала; – точно используется терминология; – показано умение иллюстрировать теоретические положения конкретными примерами, применять их в новой ситуации; – продемонстрировано усвоение ранее изученных сопутствующих вопросов; – допущены одна-две неточности при освещении второстепенных вопросов, которые исправляются по замечанию;	0-15