МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «ЛУГАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ ВЛАДИМИРА ДАЛЯ»

Факультет компьютерных систем и информационных технологий
Кафедра информатики и программной инженерии

УТВЕРЖДАЮ
Декан факультета компьютерных систем и информационных технологий
_____ Кочевский А. А.
«19» 04 2023 г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
по учебной дисциплине

«Параллельное программирование»

09.04.04 Программная инженерия

«Методы и средства разработки программного обеспечения»

Разработчик:
доцент _____ Письменский А.В.

ФОС рассмотрен и одобрен на заседании кафедры информатики и программной инженерии
от 18 апреля 2023 г., протокол № 17

Заведующий кафедрой _____ Кочевский А.А.

Луганск 2023 г.

Паспорт
фонда оценочных средств по учебной дисциплине
«Параллельное программирование»

Перечень компетенций (элементов компетенций), формируемых в результате освоения учебной дисциплины

№ п/п	Код контролируемой компетенции	Формулировка контролируемой компетенции	Контролируемые темы учебной дисциплины	Этапы формирования (семестр изучения)
1	ПК-1.1	Знать: нормативно-технические документы (стандарты и регламенты), лучшие мировые практики управления процессом разработки программного продукта	Тема 1. Тема 2.	начальный (1)
2	ПК-1.2	Уметь: применять методологии управления проектами разработки программного обеспечения	Тема 3. Тема 4.	начальный (1)
3	ПК-1.3	Владеть: навыками планирования процесса разработки программного продукта	Тема 5. Тема 6.	начальный (1)

Показатели и критерии оценивания компетенций, описание шкал оценивания

№ п/п	Код контролируемой компетенции	Показатель оценивания (знания, умения, навыки)	Контролируемые темы учебной дисциплины	Наименование оценочного средства
1.	ПК-1.1	**Знать:** методы организации потокобезопасной очереди на базе условных переменных методы доступа к результатам асинхронных операций; основные концепции библиотек работы со временем **Уметь:** проектировать параллельные и распределенные системы; использовать библиотеку	Тема 1. Синхронизация параллельных операций. Тема 2. Ожидание одноразовых событий с помощью механизма будущих результатов.	Вопросы для обсуждения (в виде докладов и сообщений), лабораторные работы, контрольные работы,

		C++ Thread Library для реализации многопоточности		
2.	ПК-1.2	**Знать:** основы функционального программирования с применением будущих результатов; атомарные типы и операции; методы синхронизация операций и принудительного упорядочения; методы упорядочения, захвата-освобождения. **Уметь:** выполнять защиту разделяемых данных с помощью условных переменных	Тема 3. Ожидание с ограничением по времени. Тема 4. Применение синхронизации операций для упрощения кода	Вопросы для обсуждения (в виде докладов и сообщений), лабораторные работы, контрольные работы,
3.	ПК-1.3	**Знать:** атомарные типы и операции; методы синхронизация операций и принудительного упорядочения; методы упорядочения **Уметь:** выполнять защиту разделяемых данных с помощью условных переменных; выполнять одновременный доступ к ресурсам; выполнять защиту разделяемых данных	Тема 5. Модель памяти C++ и атомарные операции. Тема 6. Синхронизация операций и принудительное упорядочение	Вопросы для обсуждения (в виде докладов и сообщений), лабораторные работы, контрольные работы,

Фонды оценочных средств по дисциплине «Параллельное программирование»

Лабораторная работа №1

Тема: Синхронизация действия, выполняемые в разных потоках

Цель работы: освоить механизмы синхронизации (условные переменные и будущие результаты (future))

Отчет должен включать в себя следующие разделы:

- формулировку задания;

- описание основных методов, используемых в лабораторной работе;
- результаты работы программы (в виде файла или в виде скриншота);
- анализ результатов.

Лабораторная работа №2

Тема: Моделирование одноразовых событий с помощью будущего результата

Цель работы: освоить механизмы шаблонного класса std::future, обеспечивающего механизм доступа к результатам асинхронных операций и std::shared_future<>

Отчет должен включать в себя следующие разделы:

- формулировку задания;
- описание основных методов, используемых в лабораторной работе;
- результаты работы программы (в виде файла или в виде скриншота);
- анализ результатов.

Лабораторная работа №3

Тема: Основные концепции библиотеки Chrono (C++)

Цель работы: изучение методов библиотеки chrono (namespace std::chrono)

Отчет должен включать в себя следующие разделы:

- формулировку задания;
- описание основных методов, используемых в лабораторной работе;
- результаты работы программы (в виде файла или в виде скриншота);
- анализ результатов.

Лабораторная работа №4

Тема: Алгоритма быстрой сортировки Quicksort

Цель работы: выполнить параллельную реализацию алгоритма Quicksort

Отчет должен включать в себя следующие разделы:

- формулировку задания;
- описание основных методов, используемых в лабораторной работе;
- результаты работы программы (в виде файла или в виде скриншота);
- анализ результатов.

Лабораторная работа №5

Тема: Атомарные типы и операции

Цель работы: освоить механизмы использования атомарных объектов и операций

Отчет должен включать в себя следующие разделы:

- формулировку задания;
- описание основных методов, используемых в лабораторной работе;
- результаты работы программы (в виде файла или в виде скриншота);
- анализ результатов.

Лабораторная работа №6

Тема: Синхронизация операций и принудительное упорядочение modification order

Цель работы: освоить механизмы упорядочивания атомарных объектов типов std::atomic<T> и std::atomic_flag.

Отчет должен включать в себя следующие разделы:

- формулировку задания;
- описание основных методов, используемых в лабораторной работе;
- результаты работы программы (в виде файла или в виде скриншота);
- анализ результатов.

Лабораторная работа №7

Тема: Зависимости по данным, упорядочение захват-освобождение и семантика memory_order_consume

Цель работы: освоить механизмы упорядочение захват-освобождение и семантика memory_order_consume

Отчет должен включать в себя следующие разделы:

- формулировку задания;
- описание основных методов, используемых в лабораторной работе;
- результаты работы программы (в виде файла или в виде скриншота);
- анализ результатов.

Критерии и шкала оценивания по оценочному средству лабораторная работа

Шкала оценивания (интервал баллов)	Критерий оценивания
5	Лабораторная работа выполнена на высоком уровне (правильность выполнения 90-100%)
4	Лабораторная работа выполнена на среднем уровне (правильность выполнения 75-89%)
3	Лабораторная работа выполнена на низком уровне (правильность выполнения 50-74%)
2	Лабораторная работа выполнена на неудовлетворительном уровне (правильность выполнения менее чем на 50%)

Вопросы для обсуждения (в виде докладов и сообщений):

1. Ассоциирование задачи с будущим результатом
2. Быстрая сортировка в духе ФП
3. Быстрая сортировка в духе ФП
4. Возврат значения из фоновой задачи
5. Временные интервалы
6. Использование std::promise
7. Механизм ослабленного упорядочения
8. Механизм ослабленного упорядочения
9. Не последовательно согласованное упорядочение доступа к памяти
10. Объекты и ячейки памяти
11. Ожидание события или иного условия

12. Ожидание условия с помощью условных переменных
13. Операции над std::atomic_flag
14. Операции над std::atomic<bool>
15. Операции над std::atomic<T*>: арифметика указателей
16. Операции над стандартными атомарными целочисленными типами
17. Ослабленное упорядочение
18. Ослабленное упорядочение
19. Основной шаблон класса std::atomic<>
20. Параллельная реализация Quicksort в духе ФП
21. Порядок модификации
22. Последовательно согласованное упорядочение
23. Потокобезопасная очередь на базе условных переменных
24. Свободные функции для атомарных операций
25. Синхронизация операций с помощью передачи сообщений
26. Сохранение исключения в будущем результат
27. Стандартные атомарные типы
28. Транзитивная синхронизация с помощью упорядочения захват-освобождение.
29. Упорядочение захват-освобождение
30. Функции, принимающие таймаут
31. Функциональное программирование с применением будущих результатов

Критерии и шкала оценивания по оценочному средству доклад, сообщение

Характеристика знания предмета и ответов	Зачеты
Студент глубоко и в полном объёме владеет программным материалом. Грамотно, исчерпывающе и логично его излагает в устной или письменной форме. При этом знает рекомендованную литературу, проявляет творческий подход в ответах на вопросы и правильно обосновывает принятые решения, хорошо владеет умениями и навыками при выполнении практических задач.	зачтено
Студент знает программный материал, грамотно и по сути излагает его в устной или письменной форме, допуская незначительные неточности в утверждениях, трактовках, определениях и категориях или незначительное количество ошибок. При этом владеет необходимыми умениями и навыками при выполнении практических задач.	
Студент знает только основной программный материал, допускает неточности, недостаточно чѐткие формулировки, непоследовательность в ответах, излагаемых в устной или письменной форме. При этом недостаточно владеет умениями и навыками при выполнении практических задач. Допускает до 30% ошибок в излагаемых ответах.	
Студент не знает значительной части программного материала. При этом допускает принципиальные ошибки в доказательствах, в трактовке понятий и категорий, проявляет низкую культуру знаний, не владеет основными умениями и навыками при выполнении практических задач. Студент отказывается от ответов на дополнительные вопросы.	не зачтено

Вопросы к контрольным работам:

1. Подходы к организации параллелизма.
2. Параллельный код: состояние гонки (race condition)..
3. Поддержка параллелизма в новом стандарте C++.
4. Использование мьютексов в C++.
5. Эффективность библиотеки многопоточности для C++.
6. Структурирование кода для защиты разделяемых данных.
7. Использования параллелизма в приложении.
8. Устранение проблематичных состояний гонки.
9. История многопоточности в C++.
10. Защита разделяемых данных с помощью мьютексов.
11. Платформенно-зависимые средства.
12. Интерфейс адаптера контейнера std::stack.
13. Запуск потока в C++.
14. Состояние гонки: *передача функции pop()ссылки на переменную.*
15. Запуск потоков в фоновом режиме.
16. Состояние гонки: *комбинированный вариант*.
17. Передача аргументов функции потока.
18. Взаимоблокировка: проблема и решение.
19. Ожидание завершения потока.
20. Состояние гонки: *наличия копирующего или перемещающего конструктора*.

Критерии и шкала оценивания по оценочному средству контрольная работа

Шкала оценивания (интервал баллов)	Критерий оценивания
5	Контрольная работа выполнена на высоком уровне (правильные ответы даны на 90-100% вопросов/задач)
4	Контрольная работа выполнена на среднем уровне (правильные ответы даны на 75-89% вопросов/задач)
3	Контрольная работа выполнена на низком уровне (правильные ответы даны на 50-74% вопросов/задач)
2	Контрольная работа выполнена на неудовлетворительном уровне (правильные ответы даны менее чем на 50%)

Курсовые работы

Темы курсовых работ:

- Синхронизация параллельных операций с помощью условных переменных;

- Организация исключительного доступа к данным с помощью объекта future.
- Организация потокобезопасной очереди на базе условных переменных;
- Реализация многопоточного алгоритма быстрой сортировки;
- Устранение проблематичных состояний гонки;
- Структурирование кода для защиты разделяемых данных;
- Гибкая блокировка с помощью std::unique_lock;
- Защита редко обновляемых структур данных;
- Реализация доступа к данным с помощью атомарных объектов;
- Упорядочивание данных с помощью атомарных объектов std::atomic<T> и std::atomic_flag;
- Упорядочение захват-освобождение и семантика memory_order_consume.

Методические указания к выполнению курсовой работы

Общие положения

Сегодня повсеместное использование информационных технологий стало объективной необходимостью. Спектр областей, в которых применяются информационные технологии, чрезвычайно широк.

Основными целями написания курсовой работы являются:

- закрепление знаний, полученных в процессе изучения дисциплины;
- выработка навыков творческого мышления;
- формирование профессиональных навыков, связанных с самостоятельной деятельностью будущего специалиста;
- оформление материалов (четкое, ясное, технически грамотное и качественное литературное изложение пояснительной записки и оформление графического материала работы);
- воспитание ответственности за качество принятых решений.

Цели и тематика курсовых работ

Целью курсовой работы по дисциплине является формирование у студентов углубленных профессиональных компетенций, связанных с использованием методов, алгоритмов, программных и технических средств реализации и использования прикладных интеллектуальных технологий обработки и анализа данных и процессов. Тематика курсовых работ разрабатывается студентом совместно с руководителем.

Каждая курсовая работа строго индивидуальна и ориентирована на развитие у студента определенной части профессиональных навыков и

умения творчески решать практические задачи. По результатам выполнения курсовой работы оформляется пояснительная записка.

Вид, структура, содержание и объем курсовой работы

Курсовая работа студента – заключительный этап изучения дисциплины и, как правило, основывается на обобщении выполненных студентом лабораторных работ или представляет собой индивидуальное задание по изучаемой дисциплине и подготавливается к защите в завершающий период теоретического обучения.

Пояснительная записка представляет собой текстовый документ, содержащий описания проблем, решаемых в курсовой работе, технические расчеты (расчет параметров, характеристик и экономических показателей объекта проектирования, а также взаимодействия его функциональных частей, элементов конструкций и дополнительных данных), описание проектируемого объекта, принцип его действия, обоснование принятых технических, технологических и технико- экономических решений.

Рекомендуется следующий **состав и порядок расположения материала** в пояснительной записке к курсовому проекту:

- титульный лист;
- задание на курсовую работу;
- содержание;
- введение (содержит описание состояния проблемы, актуальность, цели и задачи проекта);
- основные главы (устанавливаются руководителем с учетом специфики темы проекта);
- заключение (включает основные результаты, полученные в ходе выполнения курсовой работы, выводы и рекомендации);
- список использованных источников, в т. ч. нормативных, проектных и справочных материалов;
- приложение (при необходимости).

Задание на курсовую работу должно содержать наименование темы проекта и предусматривать по возможности комплексное решение исследовательских задач. Вместе с тем, один из частных вопросов задания того или иного характера выделяется в качестве специальной части проекта и подлежит более глубокой разработке на основе общего решения.

В задании на курсовую работу указывается:

- наименование работы;
- содержание курсовой работы и рекомендуемый объем отдельных частей;

- специальная часть проекта;
- исходные данные;
- рекомендуемая литература;
- календарный график работы студента.

Задание на курсовую работу должно содержать элемент новизны, активизирующий инициативу студента. Каждое задание должно быть индивидуальным, а его тематика, по возможности, комплексной, охватывающей несколько взаимосвязанных задач.

Возможны «сквозные» задания, отдельные аспекты которых студент выполняет в течение нескольких семестров по нескольким, следующим друг за другом, дисциплинам, и которые могут входить в состав задания на выпускную квалификационную работу.

Основные главы пояснительной записки включает следующие разделы:

1. Анализ прикладной области, приводящий к конкретной постановке задачи с выделением совокупности понятий (интенсионалы и экстенсионалы понятий), подлежащих анализу методами интеллектуального анализа.

Постановка задачи включает:

- цель разработки в конкретной прикладной области для выбранных данных;
- перечень и методы решаемых задач для конкретных данных из выбранной прикладной области;
- общие требования к данным и результатам на уровне вход/выход;
- техническое задание с выбором конкретных методов решаемых задач.

2. Проектирование:

- анализ аналогов решения обозначенных в предыдущем разделе задач;
- выбор, системное и математическое описание конкретного метода (методов) анализа для решения поставленных задач;
- описание мер точности и адекватности применяемых методов;
- проектирование и описание системы хранения данных и систем знаний;
- описание онтологии, если имеется;
- проектирование архитектуры программной системы.

4. Реализация:

- выбор инструментальных средств;
- описание характеристик программной системы;
- разработка и описание пользовательского интерфейса;

- описание используемых компонент системы;
- инструкция пользователю с копиями экранных форм.

Правила оформления пояснительной записки

Пояснительная записка к курсовой работе оформляется в соответствии с требованиями действующих государственных стандартов.

Пояснительная записка может быть оформлена в текстовом процессоре.

Гарнитура TimesNewRoman, 14 пт, оглавление, разбивка на страницы, поля: справа – 2,5 см; сверху снизу– 2см, сверху – 2см, слева – 2см.

Материал пояснительной записки излагается грамотно, четко, сжато. Пояснительная записка должна быть сброшюрована.

Каждая курсовая работа должна проходить нормоконтроль на соответствие оформления общим требованиям к текстовым документам согласно стандартам. Нормоконтроль может проводиться как руководителем курсового проекта, так и утвержденным кафедрой нормоконтролером.

Критерии и шкала оценивания по оценочному средству курсовая работа

Шкала оценивания (интервал баллов)	Критерий оценивания
5	Курсовая работа выполнена на высоком уровне (правильность выполнения 90-100%)
4	Курсовая работа выполнена на среднем уровне (правильность выполнения 75-89%)
3	Курсовая работа выполнена на низком уровне (правильность выполнения 50-74%)
2	Курсовая работа выполнена на неудовлетворительном уровне (правильность выполнения менее чем на 50%)

Оценочные средства для промежуточной аттестации (экзамен)

1. Ассоциирование задачи с будущим результатом.
2. Атомарные операции и типы в C++.
3. базе условных переменных.
4. Возврат значения из фоновой задачи.
5. Временные интервалы.
6. Выполнить асинхронные вызовы, функция std::async.
7. Выполнить параллельную реализацию алгоритма Quicksort.
8. Выполнить синхронизацию действий с использованием

потокобезопасной очереди на

9. Выполнить синхронизацию действий с помощью разделяемого флага (защищенный мьютексом).

10. Выполнить синхронизацию действий с помощью функции std::this_thread::sleep_for().

11. Выполнить синхронизацию действий средствами из стандартной библиотеки C++

12. Выполнить синхронизацию действий, используя условные переменные.

13. Гибкая блокировка с помощью std::unique_lock.

14. Дополнительные рекомендации, как избежать взаимоблокировок.

15. Задание количества потоков во время выполнения.

16. Защита разделяемых данных во время инициализации.

17. Идентификация потоков.

18. Интерфейс адаптера контейнера std::stack.

19. Интерфейс класса std::queue.

20. Интерфейс класса threadsafe_queue.

21. Использование std::future для получения результата асинхронной задачи.

22. Использование std::promise.

23. Использование иерархии блокировок для предотвращения взаимоблокировок.

24. Механизм ослабленного упорядочения.

25. Механизмы шаблонного класса std::future.

26. Модель простого конечного автомата для банкомата.

27. Моменты времени.

28. Обработка нескольких соединений в одном потоке с помощью объектов-обещаний.

29. Объекты и ячейки памяти в параллельных задачах.

30. Ожидание завершения потока в случае исключения.

31. Ожидание события или иного условия.

32. Ожидание условия с помощью условных переменных.

33. Ожидание условной переменной с таймаутом.

34. Операции над стандартными атомарными целочисленными типами.

35. Организация порядка данных и std::memory_order_seq_cst.

36. Основные концепции библиотеки Chrono.

37. Передача аргументов функции, заданной в std::async.

38. Передача владения мьютексом между контекстами.

39. Передача владения потоком.

40. Полное определение класса потокобезопасной очереди на базе условныхпеременных.

41. Порядок модификации.

42. Потокобезопасная очередь на базе условных переменных.

43. Применение std::lock() и stdt:lock_guard для реализации операции обмена.

44. Применение std::lock() и stdt:lock_guard дня реализации операции обмена.

45. Применение синхронизации операций для упрощения кода.

46. Пример использование std :: async вместо std :: thread

47. Пример использования std::atomic_flag.

48. Пример использования std::atomic<bool>.

49. Пример использования std::atomic<T*>: арифметика указателей.

50. Пример использования операции чтения-изменения-записи.

51. Пример использования функции try_lock_for(duration).

52. Пример использования функции wait_for(lock, duration, predicate).

53. Пример использования шаблонного класса std::chrono::duration.

54. Пример использования шаблонного класса std::ratio.

55. Пример использования шаблонного класса time_point.

56. Пример реализации методов std::promise::set_exception и std::promise::set_exception_at_thread_exit

57. Пример реализации шаблона класса std::chrono::time_point<>.

58. Пример упорядочивания ослабленных операций с помощью барьеров.

59. Пример чтения из очереди с применением атомарных операций.

60. Принудительное задание упорядочения неатомарных операций с помощью атомарных.

61. Проблемы разделения данных между потоками.

62. Реализация алгоритма std::partition().

63. Реализация иерархического мьютекса.

64. Реализация интерфейса с применением std::packaged_task.

65. Реализация локального, по отношению к потоку, хранилища.

66. Реализация модели захвата-освобождения.

67. Реализация модели последовательно согласованности(sequential consistency).

68. Реализация рекурсивной сортировки в духе ФП.

69. Реализация транзитивной синхронизации с помощью упорядочения захватосвобождение.

70. Свободные функции для атомарных операций, примеры.

71. Семантика перемещения.

72. Синхронизация операций с помощью передачи сообщений.

73. Синхронизация параллельных операций.

74. Состояние гонки: *возврат указатель на вытолкнутый элемент*.

75. Сохранение исключения в будущем результате.

76. Ссылки на *r*-значения и шаблоны функций.

77. Ссылки на *r*-значения.

78. Упорядочение доступа к памяти для атомарных операций.

79. Функции, принимающие таймаут.

80. Функциональное программирование с применением будущих результатов.

81. Шаблон класса std::atomic<>.

82. Ячейки памяти(memory location.

Критерии и шкала оценивания по оценочному средству промежуточный контроль (экзамен)

Шкала оценивания (интервал баллов)	Критерий оценивания
отлично (5)	Студент глубоко и в полном объёме владеет программным материалом. Грамотно, исчерпывающе и логично его излагает в устной или письменной форме. При этом знает рекомендованную литературу, проявляет творческий подход в ответах на вопросы и правильно обосновывает принятые решения, хорошо владеет умениями и навыками при выполнении практических задач.
хорошо (4)	Студент знает программный материал, грамотно и по сути излагает его в устной или письменной форме, допуская незначительные неточности в утверждениях, трактовках, определениях и категориях или незначительное количество ошибок. При этом владеет необходимыми умениями и навыками при выполнении практических задач.
удовлетворительно (3)	Студент знает только основной программный материал, допускает неточности, недостаточно чёткие формулировки, непоследовательность в ответах, излагаемых в устной или письменной форме. При этом недостаточно владеет умениями и навыками при выполнении практических задач. Допускает до 30% ошибок в излагаемых ответах.
неудовлетворительно (2)	Студент не знает значительной части программного материала. При этом допускает принципиальные ошибки в доказательствах, в трактовке понятий и категорий, проявляет низкую культуру знаний, не владеет основными умениями и навыками при выполнении практических задач. Студент отказывается от ответов на дополнительные вопросы

Лист изменений и дополнений

№ п/п	Виды дополнений и изменений	Дата и номер протокола заседания кафедры (кафедр), на котором были рассмотрены и одобрены изменения и дополнения	Подпись (с расшифровкой) заведующего кафедрой (заведующих кафедрами)

Экспертное заключение

Представленный фонд оценочных средств (далее – ФОС) по дисциплине «Параллельное программирование» соответствует требованиям ФГОС ВО.

Предлагаемые формы и средства текущего и промежуточного контроля адекватны целям и задачам реализации основной профессиональной образовательной программы по направлению подготовки 09.04.04 Программная инженерия.

Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины представлены в полном объеме.

Виды оценочных средств, включенные в представленный фонд, отвечают основным принципам формирования ФОС.

Разработанный и представленный для экспертизы фонд оценочных средств рекомендуется к использованию в процессе подготовки обучающихся по указанному направлению.

Председатель учебно-методической комиссии факультета компьютерных систем и информационных технологий — Ветрова Н. Н.