

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ  
ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Забайкальский государственный университет»  
(ФГБОУ ВО «ЗабГУ»)

Энергетический факультет

Кафедра Информатики, вычислительной техники и прикладной математики

УТВЕРЖДАЮ:

Декан факультета

Энергетический факультет

Батухтин Андрей  
Геннадьевич

«\_\_\_\_\_» \_\_\_\_\_ 20\_\_\_\_  
г.

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)**

Б1.В.ДВ.02.01 Теория вычислительных процессов  
на 144 часа(ов), 4 зачетных(ые) единиц(ы)  
для направления подготовки (специальности) 09.03.01 - Информатика и вычислительная  
техника

составлена в соответствии с ФГОС ВО, утвержденным приказом  
Министерства образования и науки Российской Федерации от  
«\_\_\_\_\_» \_\_\_\_\_ 20\_\_\_\_ г. №\_\_\_\_\_

Профиль – Программное обеспечение вычислительной техники и автоматизированных  
систем (для набора 2021)  
Форма обучения: Очная

# 1. Организационно-методический раздел

## 1.1 Цели и задачи дисциплины (модуля)

Цель изучения дисциплины:

Получение студентами знаний в области теории вычислительных процессов; о методах формальной спецификации и верификации программ; формальном описании и моделировании вычислительных процессов, их синхронизации и реализации этих подходов на языке высокого уровня.

Задачи изучения дисциплины:

- Ознакомление с методами формальной спецификации и верификации;
- Овладение методами моделирования, способами реализации и областями применения сетей Петри.
- Получения навыков параллельного программирования, синхронизации и взаимодействия процессов и потоков.

## 1.2. Место дисциплины (модуля) в структуре ОП

Дисциплина изучается на третьем курсе в шестом семестре. Для успешного освоения дисциплины требуется знания и навыки полученные при изучении дисциплин: Программирование, Объектно-ориентированное программирование, линейная алгебра и аналитическая геометрия, математическая логика и теория алгоритмов. Знания, умения и навыки могут быть востребованы на дисциплинах связанных с созданием программ, использующих параллельные алгоритмы Технологии разработки программного обеспечения, Интерактивные графические системы, при прохождении производственной практики.

## 1.3. Объем дисциплины (модуля) с указанием трудоемкости всех видов учебной работы

Общая трудоемкость дисциплины (модуля) составляет 4 зачетных(ые) единиц(ы), 144 часов.

Виды занятий	Семестр 6	Всего часов
Общая трудоемкость		144
Аудиторные занятия, в т.ч.	64	64
Лекционные (ЛК)	32	32
Практические (семинарские) (ПЗ, СЗ)	0	0
Лабораторные (ЛР)	32	32
Самостоятельная работа студентов (СРС)	80	80
Форма промежуточной	Зачет	0

аттестации в семестре		
Курсовая работа (курсовой проект) (КР, КП)		

## 2. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы

Планируемые результаты освоения образовательной программы		Планируемые результаты обучения по дисциплине
Код и наименование компетенции	Индикаторы достижения компетенции, формируемые в рамках дисциплины	Дескрипторы: знания, умения, навыки и (или) опыт деятельности
ОПК-8	<p>ОПК-8.1.Знать: основные языки программирования и работы с базами данных, операционные системы и оболочки, современные программные среды разработки информационных систем и технологий.</p> <p>ОПК-8.2.Уметь: применять языки программирования и работы с базами данных, современные программные среды разработки информационных систем и технологий для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ</p> <p>ОПК-8.3.Иметь навыки: программирования, отладки и тестирования прототипов программно-технических комплексов задач</p>	<p>Знать: Знать: основные языки программирования и работы с базами данных, операционные системы и оболочки, современные программные среды разработки информационных систем и технологий</p> <p>Уметь: Уметь: применять языки программирования и работы с базами данных, современные программные среды разработки информационных систем и технологий для автоматизации бизнес- процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ.</p> <p>Владеть: Владеть: программирования, отладки и тестирования прототипов программно-технических комплексов задач.</p>
ПК-1	<p>ПК-1.1 методологии разработки программного обеспечения, назначение и возможности средств проектирования программного обеспечения</p> <p>ПК-1.2 разрабатывать</p>	<p>Знать: Знать: методологии разработки программного обеспечения, назначение и возможности средств проектирования программного обеспечения</p>

<p>функциональные и иные требования к программным и программно-аппаратным средствам, осуществлять документирование на всех этапах проектирования и разработки, анализировать или самостоятельно разрабатывать требования к программному обеспечению; проектировать программные продукты для решения практических задач согласно разработанным требованиям; создавать программное обеспечения согласно разработанным проектам</p> <p>ПК-1.3 Иметь навыки разработки требований к программным продуктам; использования методов и средств проектирования программного обеспечения; создания программного обеспечения по разработанным проектам для решения практических и профессиональных задач.</p>	<p>Уметь: Уметь: разрабатывать функциональные и иные требования к программным и программно-аппаратным средствам, осуществлять документирование на всех этапах проектирования и разработки, анализировать или самостоятельно разрабатывать требования к программному обеспечению; проектировать программные продукты для решения практических задач согласно разработанным требованиям; создавать программное обеспечения согласно разработанным проектам</p> <p>Владеть: Владеть: разработки требований к программным продуктам; использования методов и средств проектирования программного обеспечения; создания программного обеспечения по разработанным проектам для решения практических и профессиональных задач.</p>
--	--

### 3. Содержание дисциплины

#### 3.1. Разделы дисциплины и виды занятий

##### 3.1 Структура дисциплины для очной формы обучения

Модуль	Номер раздела	Наименование раздела	Темы раздела	Всего часов	Аудиторные занятия			С Р С
					Л К	П З (С З)	Л Р	
1	1.1	Верификация программ и контрактное программирование	Понятие верификации. Виды верификации. Тройка Хоара. Формальная	22	6	0	6	10

		ние	верификация. Вывод слабейшего предусловия. Контрактное программирование. Библиотеки и средства языков программирования для контрактов.					
	1.2	Схемы программ	Схема программы. Состав схемы. Класс стандартных схем. Интерпретация схемы.	12	4	0	4	4
2	2.1	Сети Петри	Позиции, переходы, входные и выходные функции. Граф. Маркировка и выполнение сети Петри. Примеры моделей на основе сети Петри. Анализ сетей Петри. Свойства. Задача достижимости (матричный способ решения). Матрица инцидентности.	48	12	0	12	24
3	3.1	Параллельное программиров ание	Классификация параллельных систем. Закон Амдала. Виды параллельной декомпозиции. Проблемы параллельных алгоритмов.	12	4	0	4	4
	3.2	Потоки, процессы, OpenMP, MPI	Использование потоков. Паттерны проектирования параллельных алгоритмов. Синхронизация. Векторизация. OpenMP. Директивы для параллельных вычислений. Организация доступа к переменным. MPI. Общее устройство.	50	6	0	6	38

			Обмен данными. Коллективные операции. Настройка MPI кластера. Распределённые вычисления на GPU.					
Итого				144	32	0	32	80

### 3.2. Содержание разделов дисциплины

#### 3.2.1. Лекционные занятия, содержание и объем в часах

Модуль	Номер раздела	Тема	Содержание	Трудоемкость (в часах)
1	1.1	Понятие верификации. Виды верификации. Тройка Хоара. Формальная верификация. Вывод слабейшего предусловия. Контрактное программирование. Библиотеки и средства языков программирования для контрактов.	Понятие верификации. Виды верификации. Тройка Хоара. Формальная верификация. Вывод слабейшего предусловия. Контрактное программирование. Библиотеки и средства языков программирования для контрактов.	6
	1.2	Схема программы. Состав схемы. Класс стандартных схем. Интерпретация схемы.	Схема программы. Состав схемы. Класс стандартных схем. Интерпретация схемы.	4
2	2.1	Позиции, переходы, входные и выходные функции.	Позиции, переходы, входные и выходные функции. Граф. Маркировка и выполнение сети Петри. Примеры моделей на основе сети Петри. Анализ сетей Петри.	12

		<p>Граф. Маркировка и выполнение сети Петри. Примеры моделей на основе сети Петри. Анализ сетей Петри. Свойства. Задача достижимости (матричный способ решения). Матрица инцидентности.</p>	<p>Свойства. Задача достижимости (матричный способ решения). Матрица инцидентности.</p>	
3	3.1	<p>Классификация параллельных систем. Закон Амдала. Виды параллельной декомпозиции . Проблемы параллельных алгоритмов.</p>	<p>Классификация параллельных систем. Закон Амдала. Виды параллельной декомпозиции. Проблемы параллельных алгоритмов.</p>	4
	3.2	<p>Использование потоков. Паттерны проектирования параллельных алгоритмов. Синхронизация. Векторизация. OpenMP. Директивы для параллельных вычислений. Организация доступа к переменным. MPI. Общее устройство. Обмен данными.</p>	<p>Использование потоков. Паттерны проектирования параллельных алгоритмов. Синхронизация. Векторизация. OpenMP. Директивы для параллельных вычислений. Организация доступа к переменным. MPI. Общее устройство. Обмен данными. Коллективные операции. Настройка MPI кластера. Распределённые вычисления на GPU.</p>	6

		Коллективные операции. Настройка MPI кластера. Распределённые вычисления на GPU.	
--	--	--	--

### 3.2.2. Практические занятия, содержание и объем в часах

Модуль	Номер раздела	Тема	Содержание	Трудоемкость (в часах)

### 3.2.3. Лабораторные занятия, содержание и объем в часах

Модуль	Номер раздела	Тема	Содержание	Трудоемкость (в часах)
1	1.1	Понятие верификации. Виды верификации. Тройка Хоара. Формальная верификация. Вывод слабейшего предусловия. Контрактное программирование. Библиотеки и средства языков программирования для контрактов.	Понятие верификации. Виды верификации. Тройка Хоара. Формальная верификация. Вывод слабейшего предусловия. Контрактное программирование. Библиотеки и средства языков программирования для контрактов.	6
	1.2	Схема программы. Состав схемы. Класс стандартных схем. Интерпретация схемы.	Схема программы. Состав схемы. Класс стандартных схем. Интерпретация схемы.	4

2	2.1	<p>Позиции, переходы, входные и выходные функции. Граф. Маркировка и выполнение сети Петри. Примеры моделей на основе сети Петри. Анализ сетей Петри. Свойства. Задача достижимости (матричный способ решения). Матрица инцидентности.</p>	<p>Позиции, переходы, входные и выходные функции. Граф. Маркировка и выполнение сети Петри. Примеры моделей на основе сети Петри. Анализ сетей Петри. Свойства. Задача достижимости (матричный способ решения). Матрица инцидентности.</p>	12
3	3.1	<p>Классификация параллельных систем. Закон Амдала. Виды параллельной декомпозиции . Проблемы параллельных алгоритмов.</p>	<p>Классификация параллельных систем. Закон Амдала. Виды параллельной декомпозиции. Проблемы параллельных алгоритмов.</p>	4
	3.2	<p>Использование потоков. Паттерны проектирования параллельных алгоритмов. Синхронизация. Векторизация. OpenMP. Директивы для параллельных вычислений. Организация</p>	<p>Использование потоков. Паттерны проектирования параллельных алгоритмов. Синхронизация. Векторизация. OpenMP. Директивы для параллельных вычислений. Организация доступа к переменным. MPI. Общее устройство. Обмен данными. Коллективные операции. Настройка MPI кластера. Распределённые вычисления на GPU.</p>	6

		<p>доступа к переменным. MPI. Общее устройство. Обмен данными. Коллективные операции. Настройка MPI кластера. Распределённые вычисления на GPU.</p>	
--	--	---	--

### 3.3. Содержание материалов, выносимых на самостоятельное изучение

Модуль	Номер раздела	Содержание материалов, выносимого на самостоятельное изучение	Виды самостоятельной деятельности	Трудоемкость (в часах)
1	1.1	<p>Понятие верификации. Виды верификации. Тройка Хоара. Формальная верификация. Вывод слабейшего предусловия. Контрактное программирование. Библиотеки и средства языков программирования для контрактов.</p>	<p>Понятие верификации. Виды верификации. Тройка Хоара. Формальная верификация. Вывод слабейшего предусловия. Контрактное программирование. Библиотеки и средства языков программирования для контрактов.</p>	10
	1.2	<p>Схема программы. Состав схемы. Класс стандартных схем. Интерпретация схемы.</p>	<p>Схема программы. Состав схемы. Класс стандартных схем. Интерпретация схемы.</p>	4
2	2.1	<p>Позиции, переходы, входные и выходные функции. Граф. Маркировка и выполнение сети Петри. Примеры моделей на основе сети Петри. Анализ сетей Петри. Свойства. Задача</p>	<p>Позиции, переходы, входные и выходные функции. Граф. Маркировка и выполнение сети Петри. Примеры моделей на основе сети Петри. Анализ сетей Петри. Свойства. Задача достижимости (матричный</p>	24

		достижимости (матричный способ решения). Матрица инцидентности.	способ решения). Матрица инцидентности.	
3	3.1	Классификация параллельных систем. Закон Амдала. Виды параллельной декомпозиции. Проблемы параллельных алгоритмов.	Классификация параллельных систем. Закон Амдала. Виды параллельной декомпозиции. Проблемы параллельных алгоритмов.	4
	3.2	Использование потоков. Паттерны проектирования параллельных алгоритмов. Синхронизация. Векторизация. OpenMP. Директивы для параллельных вычислений. Организация доступа к переменным. MPI. Общее устройство. Обмен данными. Коллективные операции. Настройка MPI кластера. Распределённые вычисления на GPU.	Использование потоков. Паттерны проектирования параллельных алгоритмов. Синхронизация. Векторизация. OpenMP. Директивы для параллельных вычислений. Организация доступа к переменным. MPI. Общее устройство. Обмен данными. Коллективные операции. Настройка MPI кластера. Распределённые вычисления на GPU.	38

#### **4. Фонд оценочных средств для проведения текущей и промежуточной аттестации обучающихся по дисциплине**

Фонд оценочных средств текущего контроля и промежуточной аттестации по итогам освоения дисциплины представлен в приложении.

[Фонд оценочных средств](#)

#### **5. Учебно-методическое и информационное обеспечение дисциплины**

##### **5.1. Основная литература**

##### **5.1.1. Печатные издания**

1.

### 5.1.2. Издания из ЭБС

1. Малявко, А. А. Параллельное программирование на основе технологий openmp, cuda, opencl, mpi : учебное пособие для вузов / А. А. Малявко. — 3-е изд., испр. и доп. — Москва : Издательство Юрайт, 2022. — 135 с. — (Высшее образование). — ISBN 978-5-534-14116-0. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/492127> (дата обращения: 06.08.2022).  
Замятина, О. М. Инфокоммуникационные системы и сети. Основы моделирования : учебное пособие для среднего профессионального образования / О. М. Замятина. — Москва : Издательство Юрайт, 2022. — 159 с. — (Профессиональное образование). — ISBN 978-5-534-10682-4. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/495530> (дата обращения: 06.08.2022).  
1. Огнева М.В. Программирование на языке C++: практический курс [Электронный ресурс]: учеб. пособие для бакалавриата и специалитета / М.В. Огнева, Е.В. Кудрина. – Москва: Издательство Юрайт, 2018. – 335 с. – (Серия: Бакалавр и специалист). – Режим доступа: [biblio-online.ru/book/7670D7EC-AC37-4675-8EAE-DD671BC6D0E4](http://biblio-online.ru/book/7670D7EC-AC37-4675-8EAE-DD671BC6D0E4)

## 5.2. Дополнительная литература

### 5.2.1. Печатные издания

1. Моделирование параллельных процессов. Сети Петри. Мараховский В. Б., Розенблюм Л. Я., Яковлев А. В. — СПб.: Профессиональная литература, 2014. – 400 с  
Теория вычислительных процессов, методическое пособие, В. Т. Калайда, 2007 г, 130 с.

### 5.2.2. Издания из ЭБС

1.

## 5.3. Базы данных, информационно-справочные и поисковые системы

Название	Ссылка
Репозиторий с материалами дисциплины	<a href="https://github.com/ivtipm/ProcessCalculus">https://github.com/ivtipm/ProcessCalculus</a>

## 6. Перечень программного обеспечения

Программное обеспечение общего назначения: ОС Microsoft Windows, Microsoft Office, ABBYY FineReader, ESET NOD32 Smart Security Business Edition, Foxit Reader, АИБС "МегаПро".

Программное обеспечение специального назначения:

- 1) Atom
- 2) JetBrains IntelliJ IDEA
- 3) JetBrains PyCharm

4) Mozilla Firefox

5) Python

6) Visual Studio Community

## 7. Материально-техническое обеспечение дисциплины

Наименование помещений для проведения учебных занятий и для самостоятельной работы обучающихся	Оснащенность специальных помещений и помещений для самостоятельной работы
Учебные аудитории для проведения занятий лекционного типа	Состав оборудования и технических средств обучения указан в паспорте аудитории, закреплённой расписанием по факультету
Учебные аудитории для проведения лабораторных занятий	
Учебные аудитории для промежуточной аттестации	
Учебные аудитории для проведения групповых и индивидуальных консультаций	Состав оборудования и технических средств обучения указан в паспорте аудитории, закреплённой расписанием по кафедре
Учебные аудитории для текущей аттестации	

## 8. Методические рекомендации по организации изучения дисциплины

Лекционные занятия. Рекомендуется вести краткий конспект со слайдов во время занятия. Перед занятием рекомендуется повторить пройденный материал, с использованием конспектов, слайдов лекций, рекомендованных и самостоятельно найденных источников; познакомиться с планом предстоящего занятия. Подготовить вопросы по пройденным темам. Лабораторные занятия. Цель занятий – углубление и закрепление теоретических знаний, формирование у них определенных умений и навыков. Перед занятием студент должен выполнить задания. Подготовиться к защите работы, знать ответы на вопросы к заданию. Рекомендуется доработать конспект. Желательно использовать несколько источников. Самостоятельная работа. Приводит студента к получению нового знания, упорядочению и углублению имеющихся знаний, формированию у него профессиональных навыков и умений. Рекомендуется с самого начала освоения курса работать с рекомендуемыми источниками и предлагаемыми заданиями в форме подготовки к очередному аудиторному занятию, дорабатывать конспект во время самостоятельной работы. При этом актуализируются имеющиеся знания, а также создается база для усвоения нового материала, возникают вопросы, ответы на которые студент получает в аудитории. По заданиям, предлагаемым для решения на лабораторных занятиях, студент должен отчитаться до наступления сессии.

Разработчик/группа разработчиков:  
Сергей Владимирович Ветров

**Типовая программа утверждена**

Согласована с выпускающей кафедрой  
Заведующий кафедрой

\_\_\_\_\_ «\_\_\_» \_\_\_\_\_ 20\_\_\_ г.