

МИНОБРНАУКИ РОССИИ  
федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Нижегородский государственный технический университет  
им. Р.Е. Алексеева» (НГТУ)

Дзержинский политехнический институт (филиал)

УТВЕРЖДАЮ  
Директор института:  
  
А.М.Петровский  
« 29 » июня 2021 г.



**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ**  
**Б1.В.ОД.2 Многопоточное программирование**  
(индекс и наименование дисциплины по учебному плану)  
для подготовки бакалавров

Направление подготовки: 01.03.04 Прикладная математика

Направленность: Математические и компьютерные методы для современных технологий

Форма обучения: очная  
Год начала подготовки 2020

Выпускающая кафедра Автоматизация, энергетика, математика и информационные системы

Кафедра-разработчик Автоматизация, энергетика, математика и информационные системы

Объем дисциплины 180/5  
часов/з.е

Промежуточная аттестация зачет с оценкой

Разработчик: ст. преподаватель Сутырин Федор Юрьевич

« 29 » 06 2021 г.

Рабочая программа дисциплины: разработана в соответствии с Федеральным государственным образовательным стандартом высшего образования (ФГОС ВО 3++) по направлению подготовки 01.03.04 Прикладная математика, утвержденного приказом МИНОБРНАУКИ РФ от 10 января 2018 года № 11 на основании учебного плана, принятого УС ДПИ НГТУ

протокол от 25.06.21 № 10


Рабочая программа одобрена на заседании кафедры-разработчика РПД Автоматизация, энергетика, математика и информационные системы  
протокол от 28.06.21 № 8

Зав. кафедрой к.т.н, доцент

 Л.Ю. Вадова  
(подпись)

СОГЛАСОВАНО:

Заведующий выпускающей кафедрой Автоматизация, энергетика, математика и информационные системы  
к.т.н, доцент

 Л.Ю. Вадова  
(подпись)

Начальник ОУМБО

 И.В. Старикова  
(подпись)

Рабочая программа зарегистрирована в ОУМБО:

Б1.В.04.2/20 ПМ

« 29 » 06 2021 г.

## СОДЕРЖАНИЕ

1. Цели и задачи освоения дисциплины .....	4
1.1. Цель освоения дисциплины.....	4
1.2. Задачи освоения дисциплины (модуля).....	4
2. Место дисциплины в структуре образовательной программы.....	4
3. Компетенции обучающегося, формируемые в результате освоения дисциплины (модуля) .....	4
4. Структура и содержание дисциплины.....	6
4.1. Распределение трудоемкости дисциплины по видам работ по семестрам.....	6
4.2. Содержание дисциплины, структурированное по темам .....	6
5. Текущий контроль успеваемости и промежуточная аттестация по итогам освоения дисциплины.....	14
5.1. Типовые контрольные задания и иные материалы, необходимые для оценки знаний, умений и навыков и (или) опыта деятельности .....	14
5.2. Описание показателей и критериев контроля успеваемости, описание шкал оценивания.....	18
6. Учебно-методическое обеспечение дисциплины.....	20
6.1. Учебная литература .....	20
6.2. Методические указания, рекомендации и другие материалы к занятиям .....	20
7. Информационное обеспечение дисциплины.....	20
7.1. Перечень информационных справочных систем .....	21
7.2. Перечень лицензионного и свободно распространяемого программного обеспечения, в том числе отечественного производства необходимого для освоения дисциплины.....	21
8. Образовательные ресурсы для инвалидов и лиц с ОВЗ.....	22
9. Материально-техническое обеспечение, необходимое для осуществления образовательного процесса по дисциплине.....	22
10. Методические рекомендации обучающимся по освоению дисциплины.....	23
10.1. Общие методические рекомендации для обучающихся по освоению дисциплины, образовательные технологии .....	23
10.2. Методические указания для занятий лекционного типа .....	24
10.3. Методические указания по освоению дисциплины на лабораторных работах .....	25
10.4. Методические указания по самостоятельной работе обучающихся .....	25
11. Оценочные средства для контроля освоения дисциплины.....	25
11.1. Типовые контрольные задания и иные материалы, необходимые для оценки знаний, умений и (или) опыта в ходе текущего контроля успеваемости.....	25
11.1.1. Типовые задания для лабораторных работ .....	25
11.1.2. Типовые задания для самостоятельной работы обучающихся очной формы.....	26
11.2. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта в ходе промежуточной аттестации по дисциплине .....	28

## 1 ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

### Цель освоения дисциплины:

Целью освоения дисциплины является освоение методов многопоточного программирования для решения задач прикладного характера с обеспечением требований информационной безопасности.

### Задачи освоения дисциплины:

— Применение алгоритмов и технологий параллельного программирования на основе библиотек классов для многопоточного программирования;

— Знание методов и свойств классов, используемых в многопоточном программировании и возможных сфер их приложений при решении практических задач.

## 2 МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Учебная дисциплина Многопоточное программирование включена в перечень дисциплин вариативной части (формируемой участниками образовательных отношений), определяющий направленность ОП. Дисциплина реализуется в соответствии с требованиями ФГОС, ОП ВО и УП.

Дисциплина базируется на следующих дисциплинах: Объектно-ориентированное программирование, алгоритмы и структуры данных.

Дисциплина Многопоточное программирование является основополагающей для выполнения выпускной квалификационной работы.

Рабочая программа дисциплины «Многопоточное программирование» для инвалидов и лиц с ограниченными возможностями здоровья разрабатывается индивидуально с учетом особенностей психофизического развития, индивидуальных возможностей и состояния здоровья таких обучающихся, по их личному заявлению.

## 3 КОМПЕТЕНЦИИ ОБУЧАЮЩЕГОСЯ, ФОРМИРУЕМЫЕ В РЕЗУЛЬТАТЕ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)

Таблица 1 – Формирование компетенции ПКС-1 дисциплинами

Наименование дисциплин, формирующих компетенцию совместно	Семестры формирования компетенций дисциплинами.							
	1	2	3	4	5	6	7	8
Объектно-ориентированное программирование								
Ознакомительная практика								
Технологии программирования								
Проектно-технологическая практика								
Многопоточное программирование								
Программирование для Интернет								
Архитектура ЭВМ и системное ПО								
Преддипломная практика								
Выполнение и защита выпускной квалификационной работы								

ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ, СООТНЕСЕННЫХ С ПЛАНИРУЕМЫМИ РЕЗУЛЬТАТАМИ ОСВОЕНИЯ ОП

Таблица 2

Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения

Код и наименование компетенции	Код и наименование индикатора достижения компетенции	Планируемые результаты обучения по дисциплине			Оценочные средства	
					Текущего контроля	Промежуточной аттестации
<b>ПКС-1</b> способен проектировать и разрабатывать компоненты программного обеспечения на основе современных парадигм, технологий и языков программирования	<b>ИПКС-1.1.</b> Использует методы и инструменты современных объектно-ориентированных языков программирования для разработки компонентов программного обеспечения.	<b>Знать:</b> основные понятия параллельного программирования, структуру, методы и свойства классов, используемых в многопоточном программировании, возможные сферы их приложений при решении практических задач, основы построения программ для систем с общей и распределенной памятью.	<b>Уметь:</b> разрабатывать программы для решения задач прикладного характера с обеспечением требований информационной безопасности.	<b>Владеть:</b> алгоритмами и технологиями параллельного программирования на основе библиотек классов для многопоточного программирования среды .NET Framework (Thread, Task, Parallel), методами отладки и тестирования.	Собеседование и отчеты при сдаче лабораторных работ	Вопросы для устного собеседования: билеты (20 билетов)

## 4 СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

### 4.1 Распределение трудоёмкости дисциплины по видам работ по семестрам

Общая трудоёмкость дисциплины составляет 5 зач.ед./180 часов, распределение часов по видам работ семестрам представлено в табл. 3 и 4.

Таблица 3 – Распределение трудоёмкости дисциплины по видам работ по семестрам

Для очного обучения

Вид учебной работы	Всего часов	Семестр
		7
<b>1. Контактная работа обучающихся с преподавателем (по видам учебных занятий) (всего), в том числе:</b>	89	89
<b>1.1. Аудиторные занятия (всего), в том числе:</b>	85	85
- лекции (Л)	34	34
- лабораторные работы (ЛР)	34	34
- практические занятия (ПЗ)	17	17
- практикумы (П)	-	-
<b>1.2. Внеаудиторные занятия (всего), в том числе:</b>	4	4
- групповые консультации по дисциплине	4	4
- групповые консультации по промежуточной аттестации (экзамен)	-	-
- индивидуальная работа преподавателя с обучающимся: - по проектированию: проект (работа) - по выполнению РГР - по выполнению КР - по составлению реферата (доклада, эссе)	-	-
<b>2. Самостоятельная работа студента (СРС) (всего)</b>	91	91
<b>Вид промежуточной аттестации зачет с оценкой</b>	зачет с оценкой	зачет с оценкой
<b>Общая трудоемкость, часы/зачетные единицы</b>	180/5	180/5

### 4.2 Содержание дисциплины, структурированное по темам

Содержание дисциплины, структурированное по темам, приведено в таблице 4.

В столбце «Вид СР» введены следующие сокращения:

«**Лекции**» – предполагает изучение материалов учебников и учебных пособий для подготовки к лекциям и повторение материала после прослушивания лекции для участия в обсуждениях на практических занятиях.

«**Практика**» - предполагает использование методических разработок для помощи при решении индивидуальных задач и решение задач из задачников.

«**Лаб.раб.**» - предполагает использование методических разработок для помощи при выполнении лабораторных работ и оформлении отчетов.

## 4.2. Содержание дисциплины, структурированное по темам

Таблица 4 – Содержание дисциплины, структурированное по темам для обучающихся очной формы обучения <sup>1</sup>

Планируемые (контролируемые) результаты освоения: код УК;ОПК;ПК и индикаторы достижения компетенций	Наименование разделов, тем	Виды учебной работы				Вид СРС	Наименование используемых активных и интерактивных образовательных технологий	Реализация в рамках Практической подготовки (трудоемкость в часах)	Наименование разработанного Электронного курса (трудоемкость в часах)
		Контактная работа			Самостоятельная работа обучающихся (СРС), час				
		Лекции, час	Лабораторные работы, час	Практические занятия, час					
<b>7 семестр</b>									
ПКС-1 ИПКС-1.1	<b>Раздел 1. Основы многопоточной обработки. Методы и свойства класса Thread</b>								Конспект лекций
	<b>Тема 1.1.</b> Основы многопоточной обработки.	1			2	<b>Лекции:</b> (6.1.1: С: 10-15)			
	<b>Тема 1.2.</b> Методы класса Thread.	2			2				
	<b>Практическое занятие №1.</b> Использование методов класса Thread для создания и запуска на исполнение потока			1	2	<b>Практика:</b> (6.1.1: С: 61-70)			
	<b>Тема 1.3.</b> Момент завершения работы потока.	1			2	<b>Лекции:</b> (6.1.1: С: 49-55)			
	<b>Практическое занятие №2.</b> Определение момента завершения потока			1	2	<b>Практика:</b> (6.1.1: С: 55-60)			
	<b>Тема 1.4.</b> Состояние потока.	1			2	<b>Лекции:</b> (6.1.1: С: 107-112)			

Планируемые (контролируемые) результаты освоения: код УК;ОПК;ПК и индикаторы достижения компетенций	Наименование разделов, тем	Виды учебной работы				Вид СРС	Наименование используемых активных и интерактивных образовательных технологий	Реализация в рамках Практической подготовки (трудоемкость в часах)	Наименование разработанного Электронного курса (трудоемкость в часах)
		Контактная работа			Самостоятельная работа обучающихся (СРС), час				
		Лекции, час	Лабораторные работы, час	Практические занятия, час					
	<b>Практическое занятие №3.</b> Приостановка, возобновление и завершение работы потока			1	2	<b>Практика:</b> (6.1.1: С: 112-120)			
	<b>Тема 1.5.</b> Приоритеты потока.	2			2	<b>Лекции:</b> (6.1.1: С: 134-140)			
	<b>Практическое занятие №4.</b> Состояния и приоритеты потоков			1	2	<b>Практика:</b> (6.1.1: С: 140-150)			
	<b>Лабораторная работа №1.</b> Программная реализация работы нескольких потоков		6			<b>Лаб.Раб.:</b> (6.1.1: С: 307 – 308)			
	<b>Итого по 1 разделу</b>	7	6	4	18				
ПКС-1 ИПКС-1.1	<b>Раздел 2.</b> Классы, реализующие механизм синхронизации, их свойства и методы								
	<b>Тема 2.1.</b> Ключевое слово lock.	1			1	<b>Лекции:</b> (6.1.1: С: 180-185)			
	<b>Практическое занятие №5.</b> Использование ключевого слова lock для синхронизации работы нескольких потоков			1	2	<b>Практика:</b> (6.1.1: С: 180-185)			
	<b>Тема 2.2.</b> Класс Monitor.	2			1	<b>Лекции:</b> (6.1.1: С: 185-190)			
	<b>Практическое занятие №6.</b> Использование класса Monitor для			1	1	<b>Практика:</b> (6.1.1: С: 185-190)			



Планируемые (контролируемые) результаты освоения: код УК;ОПК;ПК и индикаторы достижения компетенций	Наименование разделов, тем	Виды учебной работы				Вид СРС	Наименование используемых активных и интерактивных образовательных технологий	Реализация в рамках Практической подготовки (трудоемкость в часах)	Наименование разработанного Электронного курса (трудоемкость в часах)
		Контактная работа			Самостоятельная работа обучающихся (СРС), час				
		Лекции, час	Лабораторные работы, час	Практические занятия, час					
	синхронизации работы нескольких потоков								
ПКС-1 ИПКС-1.1	<b>Тема 2.4.</b> Класс Mutex.	1			1	<b>Лекции:</b> (6.1.2: С: 74-77)			
	<b>Практическое занятие №7.</b> Использование классов Mutex и Semaphore. Для синхронизации работы нескольких потоков			1	1	<b>Практика:</b> (6.1.2: С: 74-77)			
	<b>Тема 2.5.</b> Класс Semaphore.	1			1	<b>Лекции:</b> (6.1.2: С: 91-93)			
	<b>Тема 2.6.</b> Класс Interlocked.	1			1	<b>Лекции:</b> (6.1.2: С: 94-96)			
	<b>Тема 2.7.</b> Класс ManualResetEvent.	1			1	<b>Лекции:</b> (6.1.2: С: 96-99)			
	<b>Тема 2.8.</b> Класс AutoResetEvent.	1			1	<b>Лекции:</b> (6.1.2: С: 99-102)			
	<b>Практическое занятие №8.</b> Использование классов ManualResetEvent и AutoResetEvent для синхронизации работы нескольких потоков			1	2	<b>Практика:</b> (6.1.2: С: 91-102)			
	<b>Лабораторная работа №2.</b> Программная реализация синхронизации нескольких потоков с помощью классов Mutex и Semaphore		6		4	<b>Лаб.Раб.:</b> (6.1.1: С: 308 – 309)			

Планируемые (контролируемые) результаты освоения: код УК;ОПК;ПК и индикаторы достижения компетенций	Наименование разделов, тем	Виды учебной работы				Вид СРС	Наименование используемых активных и интерактивных образовательных технологий	Реализация в рамках Практической подготовки (трудоемкость в часах)	Наименование разработанного Электронного курса (трудоемкость в часах)
		Контактная работа			Самостоятельная работа обучающихся (СРС), час				
		Лекции, час	Лабораторные работы, час	Практические занятия, час					
	<b>Итого по 2 разделу</b>	8	6	4	17				
ПКС-1 ИПКС-1.1	<b>Раздел 3.</b> Классы, реализующие механизм синхронизации, представленные в среде .NET Framework версии 4.0 и выше								
	<b>Тема 3.1.</b> Структура SpinLock.	2			2	<b>Лекции:</b> (6.1.2: С: 125-130)			
	<b>Практическое занятие №9.</b> Использование структуры SpinLock. Для синхронизации работы нескольких потоков.			1	2	<b>Практика:</b> (6.1.2: С: 125-130)			
	<b>Тема 3.2.</b> Класс Barrier.	2			1	<b>Лекции:</b> (6.1.2: С: 130-133)			
	<b>Практическое занятие №10.</b> Использование класса Barrier. Для синхронизации работы нескольких потоков			1	2	<b>Практика:</b> (6.1.2: С: 130-133)			
	<b>Тема 3.3.</b> Класс ReaderWriterLockSlim	2			5	<b>Лекции:</b> (6.1.2: С: 133-138)			
	<b>Практическое занятие №11.</b> Использование класса ReaderWriterLockSlim. Для синхронизации работы нескольких потоков.			1	2	<b>Практика:</b> (6.1.2: С: 133-138)			
	<b>Лабораторная работа №3.</b> Программная реализация синхронизации нескольких потоков с		8		4	<b>Лаб.Раб.:</b> (6.1.1: С. 309 – 310)			

Планируемые (контролируемые) результаты освоения: код УК;ОПК;ПК и индикаторы достижения компетенций	Наименование разделов, тем	Виды учебной работы				Вид СРС	Наименование используемых активных и интерактивных образовательных технологий	Реализация в рамках Практической подготовки (трудоемкость в часах)	Наименование разработанного Электронного курса (трудоемкость в часах)
		Контактная работа			Самостоятельная работа обучающихся (СРС), час				
		Лекции, час	Лабораторные работы, час	Практические занятия, час					
	помощью классов, реализованных в среде .NET Framework версии 4.0 и выше								
	<b>Итого по 3 разделу</b>	6	8	3	18				
ПКС-1 ИПКС-1.1	<b>Раздел 4. Библиотека распараллеливания задач (TPL)</b>								
	<b>Тема 4.1.</b> Параллельное программирование Параллелизм данных	2			1	<b>Лекции:</b> (6.1.1: С: 61-63)			
	<b>Тема 4.2.</b> Отмена цикла Parallel.For или Parallel.ForEach	1			1	<b>Лекции:</b> (6.1.1: С: 65-69)			
	<b>Практическое занятие №12.</b> Использование класса Parallel, его методов и свойств			1	2	<b>Практика:</b> (6.1.1: С: 61-69)			
	<b>Тема 4.3.</b> Обработка исключений и повышение быстродействия программы	2			2	<b>Лекции:</b> (6.1.1: С: 134-137)			
	<b>Практическое занятие №13.</b> Использование класса ThreadPool			1	2	<b>Практика:</b> (6.1.1: С: 180-185)			
	<b>Практическое занятие №14.</b> Использование структуры CancellationToken			1	2	<b>Практика:</b> (6.1.1: С: 186-190)			
	<b>Лабораторная работа №4.</b> Программная реализация параллельного выполнения задач программы с помощью класса Parallel		7		6	<b>Лаб.Раб.:</b> (6.1.2: С: 161-163)			

Планируемые (контролируемые) результаты освоения: код УК;ОПК;ПК и индикаторы достижения компетенций	Наименование разделов, тем	Виды учебной работы				Вид СРС	Наименование используемых активных и интерактивных образовательных технологий	Реализация в рамках Практической подготовки (трудоемкость в часах)	Наименование разработанного Электронного курса (трудоемкость в часах)
		Контактная работа			Самостоятельная работа обучающихся (СРС), час				
		Лекции, час	Лабораторные работы, час	Практические занятия, час					
	<b>Итого по 4 разделу</b>	5	7	3	16				
ПКС-1 ИПКС-1.1	<b>Раздел 5. Параллельный язык интегрированных запросов (PLINQ)</b>								
	<b>Тема 5.1.</b> Введение в PLINQ	2			2	<b>Лекции:</b> (6.1.3: С: 247-255)			
	<b>Тема 5.2.</b> Класс ParallelEnumerable	2			2	<b>Лекции:</b> (6.1.4: С: 6-9)			
	<b>Практическое занятие №15.</b> Использование ParallelEnumerable класса			2	2	<b>Практика:</b> (6.1.4: С: 6-9)			
	<b>Тема 5.3.</b> Операторы запросов и упорядочение	2			2	<b>Лекции:</b> (6.1.1: С: 249-252)			
	<b>Практическое занятие №16.</b> Использование операторов запросов			1	2	<b>Лекции:</b> (6.1.1: С: 252-260)			
	<b>Тема 5.4.</b> Режимы выполнения запросов	2			2	<b>Лекции:</b> (6.1.1: С: 260-278)			
	<b>Лабораторная работа №5.</b> Программная реализация параллельного выполнения программы с помощью PLINQ.		7		7	<b>Лаб.Раб.:</b> (6.1.4: С: 29-34)			
	<b>Итого по 5 разделу</b>	8	7	3	19				
	<b>Самостоятельная работа по подготовке к зачёту</b>				3				

Планируемые (контролируемые) результаты освоения: код УК;ОПК;ПК и индикаторы достижения компетенций	Наименование разделов, тем	Виды учебной работы				Вид СРС	Наименование используемых активных и интерактивных образователь- ных технологий	Реализация в рамках Практической подготовки (трудоемкость в часах)	Наименование разработанного Электронного курса (трудоемкость в часах)
		Контактная работа			Самостоятельная работа обучающихся (СРС), час				
		Лекции, час	Лабораторные работы, час	Практические занятия, час					
	<b>ИТОГО ЗА СЕМЕСТР</b>	<b>34</b>	<b>34</b>	<b>17</b>	<b>91</b>				
	<b>ИТОГО по дисциплине</b>	<b>34</b>	<b>34</b>	<b>17</b>	<b>91</b>				

## 5 ТЕКУЩИЙ КОНТРОЛЬ УСПЕВАЕМОСТИ И ПРОМЕЖУТОЧНАЯ АТТЕСТАЦИЯ ПО ИТОГАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ

### 5.1. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений и навыков и (или) опыта деятельности

#### 1) Вопросы для собеседования перед лабораторными работами:

- 1) Какой метод класса Thread позволяет приостановить выполнение потока? – Suspend ( )
- 2) Какой метод класса Thread позволяет возобновить выполнение потока? – Resume ( )
- 3) Какой метод класса Thread позволяет завершить выполнение потока? – Abort ( )
- 4) Если вызывающий поток находится не в подходящем для вызываемого метода состоянии, то какой тип исключения сгенерируется? – ThreadStateException
- 5) Какой тип исключения сгенерируется, если будет вызван метод Abort ( ) для потока, работа которого уже была завершена? – ThreadAbortException
- 6) Выберите все форматы определения метода Abort ( ). – public void Abort (object info) и public void Abort ( )
- 7) Какое свойство класса ThreadAbortException содержит информацию о завершающемся с помощью метода Abort ( ) потоке? – ExceptionState
- 8) Какие из приведенных методов являются устаревшими для среды .NET Framework? – Suspend ( ) и Resume ( ).
- 9) Какой формат определения имеет метод Suspend ( )? – public void Suspend ( )
- 10) Какой формат определения имеет метод Resume ( ) ? – public void Resume ( )
- 11) Какое свойство класса Thread позволяет отследить момент завершения работы потока? – IsAlive ( )
- 12) Какой метод класса Thread позволяет отследить момент завершения работы потока? – Join ( )
- 13) С помощью какого свойства класса Thread можно поменять тип потока? – IsBackground
- 14) На что влияет смена приоритета потока? – быстрдействие его выполнения и его взаимодействие с другими потоками
- 15) Целью назначения приоритета потокам является? – распределение процессорного времени между потоков
- 16) Какое перечисление класса Thread позволяет сменить приоритета потока? – Priority
- 17) Сколько приоритетов содержит свойство Priority класса Thread? – 5
- 18) Какой приоритет назначается по умолчанию при создании потока? – Priority.Normal
- 19) Какой наименьший приоритет реализован в свойстве Priority? – ThreadPriority.Lowest
- 20) Какой наивысший приоритет реализован в свойстве Priority? – ThreadPriority.Highest
- 21) Сколько существует разновидностей многозадачности? – 2
- 22) Какое минимальное количество потоков должен содержать каждый процесс? – 1
- 23) Количество состояний потока? – 6
- 24) Какую область виртуальной памяти используют потоки одного процесса? (общую или отдельную) – общую
- 25) Какую область виртуальной памяти используют два процесса? – отдельную
- 26) Наименьшая единица кода при организации многозадачности на основе процесса? – программа
- 27) Сколько разновидностей потока, реализованных в среде .NET Framework? – 2
- 28) Каким становится поток при его создании (по умолчанию)? – приоритетным
- 29) В каком пространстве имен среды .NET Framework реализована многопоточность? – System.Threading
- 30) Какой вид конструктора реализован в классе Thread? – public Thread (ThreadStart entry\_method)
- 31) Каким должен быть метод, передаваемый в качестве аргумента в конструктор класса Thread? – тип возвращаемого значения: void, не принимать аргументов
- 32) Какой метод запускает поток на выполнение? – метод Start ( )
- 33) Что происходит с потоком при вызове метода Sleep ( ) ? – поток приостанавливает свое выполнение на заданное количество миллисекунд
- 34) Отметьте основные преимущества использования многопоточности? – экономия времени простоя программы, одновременное решение двух и более задач

- 35) Причины завершения работы потока? – отработал метод, с которого стартовал поток; завершил работу приоритетный поток, вызов метода Abort ( )
- 36) Какое ключевое слово реализует механизм синхронизации в среде .NET Framework? – lock
- 37) Укажите правильное определение lock-конструкции? – lock ( Object ) или lock ( Object ) { }
- 38) Что необходимо передать в качестве аргумента lock-конструкции? – переменную ссылочного типа объекта блокировки
- 39) Какой метод класса Monitor блокирует объект (или общий ресурс) на время выполнения работы потока? – Enter ( )
- 40) Какой формат определения имеет метод Enter ( )? – public static void Enter ( object syncObj )
- 41) Что необходимо передать в качестве аргумента в метод Enter ( ) класса Monitor? – переменную ссылочного типа объекта блокировки
- 42) Какой метод класса Monitor снимает блокировку с объекта (или общего ресурса) для выполнения работы другого потока? – Exit ( )
- 43) Какой формат определения имеет метод Exit? – public static void Exit ( object syncObj )
- 44) Что необходимо передать в качестве аргумента в метод Exit класса Monitor? – переменную ссылочного типа объекта блокировки
- 45) Какой метод класса Monitor имеет возможность отслеживать возможность получить блокировку объект (или общий ресурс) на время выполнения работы потока? – TryEnter ( )
- 46) Что необходимо передать в качестве аргумента в метод TryEnter класса Monitor? – переменную ссылочного типа объекта блокировки
- 47) К каким методам происходит неявное обращение при использовании ключевого слова lock? – методы Enter и Exit класса Monitor
- 48) В какую из нижеперечисленных конструкций компилятор преобразует ключевое слово lock? – Monitor.Enter (lockObj);  
try  
{  
}  
finally  
{  
Monitor.Exit (lockObj);  
}
- 49) Какой атрибут синхронизирует работу всего класса? – [Synchronization]
- 50) Какой атрибут синхронизирует работу всего метода? – [MethodImplAttribute (MethodImplOptions.Synchronized)]
- 51) Какой метод структуры SpinLock блокирует объект (или общий ресурс) на время выполнения работы потока? – Enter ( )

## **7) ВОПРОСЫ к зачету с оценкой за 7-й семестр**

1. Какой метод структуры SpinLock снимает блокировку с объекта (или общего ресурса) для выполнения работы другого потока? – Exit ( )
2. Какой метод структуры SpinLock имеет возможность отслеживать возможность получить блокировку объект (или общий ресурс) на время выполнения работы потока? – TryEnter ( )
3. Укажите форму определения структуры SpinLock? – public struct SpinLock
4. С помощью какого свойства структуры SpinLock можно определить, имеет ли какой-либо поток блокировку в настоящий момент? – IsHeld
5. С помощью какого свойства структуры SpinLock можно определить, имеет ли текущий поток блокировку в настоящий момент? – IsHeldByCurrentThread
6. Укажите наиболее употребляемые формы определения класса Barrier? – public Barrier (int participantCount) и public Barrier (int participantCount, Action<System.Threading.Barrier> postPhaseAction)

7. Какие значения передаются в качестве аргумента при создании экземпляра класса `Barrier`? – Количество участвующих потоков и Действие, выполняемое после каждой фазы
8. С помощью какого свойства класса `Barrier` можно получить номер текущей фазы барьера? – `CurrentPhaseNumber`
9. С помощью какого свойства класса `Barrier` можно получить общее количество участников? – `ParticipantCount`
10. С помощью какого свойства класса `Barrier` можно получить общее количество участников, которые еще не создали сигнал в текущей фазе? – `ParticipantsRemaining`
11. Какой метод уведомляет `Barrier` о добавлении дополнительного участника? – `AddParticipant ()`
12. Укажите форму определения метода `AddParticipant` класса `Barrier`? – `public long AddParticipant ()`
13. Какой метод уведомляет `Barrier` о добавлении нескольких дополнительных участников? – `AddParticipants ()`
14. Укажите форму определения метода `AddParticipants` класса `Barrier`? – `public long AddParticipants (int participantCount)`
15. Какой метод уведомляет `Barrier` об удалении одного участника? – `RemoveParticipant`
16. Укажите форму определения метода `RemoveParticipant`? – `public void RemoveParticipant ()`
17. Какой метод уведомляет `Barrier` об удалении нескольких участников? – `RemoveParticipants`
18. Укажите форму определения метода `RemoveParticipants`? – `public void RemoveParticipants (int participantCount)`
19. С помощью какого метода участник может сообщить, что достиг барьера и ожидает достижения барьера другими участниками? – `SignalAndWait`
20. Укажите правильную форму объявления конструктора класса `ManualResetEvent`? – `public ManualResetEvent (bool initialState)`
21. Укажите правильную форму объявления конструктора класса `AutoResetEvent`? – `public AutoResetEvent (bool initialState)`
22. Какой аргумент необходимо передать в конструктор класса `ManualResetEvent`? – логическое значение `true` или `false`
23. Какой аргумент необходимо передать в конструктор класса `AutoResetEvent`? – логическое значение `true` или `false`
24. С помощью какого метода класса `ManualResetEvent` выполнение вызывающего потока приостанавливается до тех пор, пока не будет получено уведомление о событии? – `WaitOne`
25. С помощью какого метода класса `ManualResetEvent` поток уведомляется о наступлении некоторого события? – `Set ()`
26. С помощью какого метода класса `ManualResetEvent` событийный объект возвращается в несигнальное состояние? – `Reset ()`
27. Укажите наиболее употребляемую форму определения метода `Set` класса `ManualResetEvent`? – `public bool Set ()`
28. Укажите наиболее употребляемую форму определения метода `WaitOne` класса `ManualResetEvent`? – `public virtual bool WaitOne ()`
29. Укажите наиболее употребляемую форму определения метода `Reset` класса `ManualResetEvent`? – `public bool Reset ()`
30. С помощью какого метода класса `AutoResetEvent` выполнение вызывающего потока приостанавливается до тех пор, пока не будет получено уведомление о событии? – `WaitOne`
31. С помощью какого метода класса `AutoResetEvent` поток уведомляется о наступлении некоторого события? – `Set ()`
32. Укажите наиболее употребляемую форму определения метода `Set` класса `AutoResetEvent`? – `public bool Set ()`
33. Укажите наиболее употребляемую форму определения метода `WaitOne` класса `AutoResetEvent`? – `public virtual bool WaitOne ()`
34. В чем принципиальное отличие использования класса `ManualResetEvent` от класса `AutoResetEvent`? – для события типа `AutoResetEvent` событийный объект автоматически переходит в несигнальное



- состояние, как только поток, ожидающий это событие, получит уведомление о нем и возобновит свое выполнение
35. Укажите наиболее употребляемые формы определения класса `Mutex`? – `public Mutex()` и `public Mutex(bool initiallyOwned)`
  36. Какой аргумент передается в конструктор класса `Mutex`? – логическая переменная, указывающая на изначальную блокировку Мьютекса вызывающим потоком
  37. С помощью какого метода класса `Mutex` блокируется доступ к общему ресурсу или объекту другим потокам? – `WaitOne`
  38. Укажите наиболее употребляемую форму определения метода `WaitOne` класса `Mutex`? – `public virtual bool WaitOne ();`
  39. С помощью какого метода класса `Mutex` освобождается ранее заблокированный объект? – `ReleaseMutex`
  40. Укажите наиболее употребляемую форму определения метода `ReleaseMutex` класса `Mutex`? – `public void ReleaseMutex ()`
  41. В чем выгодное отличие класса `Semaphore` от класса `Mutex`? – предоставляет одновременный доступ к общему ресурсу не одному, а нескольким потокам
  42. Укажите наиболее употребляемую форму определения класса `Semaphore`? – `public Semaphore(int initialCount, int maximumCount)`
  43. Какие аргументы передаются в конструктор класса `Semaphore`? – количество первоначально доступных разрешений и максимальное количество разрешений
  44. С помощью какого метода класса `Semaphore` блокируется доступ к общему ресурсу или объекту другим потокам? – `WaitOne`
  45. Укажите наиболее употребляемую форму определения метода `WaitOne` класса `Semaphore`? – `public virtual bool WaitOne ();`
  46. С помощью какого метода класса `Semaphore` освобождается ранее заблокированный объект? – `Release`
  47. Укажите наиболее употребляемую форму определения метода `Release` класса `Semaphore`? – `public int Release ()` и `public int Release (int releaseCount)`
  48. Какое значение необходимо передать в качестве аргумента в метод `Release` класса `Semaphore`? – количество разрешений для объекта Семафор
  49. С помощью какого метода класса `Monitor` поток переходит в режим ожидания разблокировки объекта (или общего ресурса) на время выполнения работы другого потока? – `Wait ()`
  50. Укажите наиболее употребляемые формы определения метода `Wait`? – `public static void Wait (object waitObj)` и `public static void Wait (object waitObj, int milliseconds)`
  51. Что необходимо передать в качестве аргумента в метод `Wait ()` класса `Monitor`? – переменную ссылочного типа, к которой ожидается доступ
  52. С помощью каких методов класса `Monitor` возобновляется выполнение работы потока? – `Pulse ()` и `PulseAll ()`
  53. Какой метод сообщает ожидающему потоку, стоящему первым в очереди потоков, о снятии блокировки с объекта? – `Pulse ()`
  54. Какой метод сообщает всем ожидающим потокам о снятии блокировки с объекта? – `PulseAll ()`
  55. Укажите наиболее употребляемую форму определения метода `Pulse ()`? – `public static void Pulse (object waitObj)`
  56. Укажите наиболее употребляемую форму определения метода `PulseAll ()`? – `public static void Pulse (object waitObj)`
  57. Что необходимо передать в качестве аргумента в метод `Pulse ()` класса `Monitor`? – переменную ссылочного типа, освобождаемую от блокировки
  58. Что необходимо передать в качестве аргумента в метод `PulseAll ()` класса `Monitor`? – переменную ссылочного типа, освобождаемую от блокировки
  59. Методы `Wait`, `Pulse` и `PulseAll` должны вызываться внутри какого блока кода? – `lock`
  60. Какое исключение сгенерируется, если методы `Wait`, `Pulse` и `PulseAll` будут вызваны вне `lock`-блока кода? – `SynchronizationLockException`

61. С помощью какого метода можно отменить завершение работы потока, который сгенерировал исключение типа `ThreadAbortException`? – `ResetAbort`
62. Какой формат определения у метода `ResetAbort`? – `public void ResetAbort ( )`
63. Какое свойство позволяет определить состояние потока? – `ThreadState`
64. Напишите значение свойства `ThreadState`, которое сигнализирует об остановке потока, но не окончательном его завершении? – `ThreadState.Aborted`
65. Напишите значение свойства `ThreadState`, которое сигнализирует о том, что поток выполняется в фоновом режиме? – `ThreadState.Background`
66. Напишите значение свойства `ThreadState`, которое сигнализирует о том, что поток завершен? – `ThreadState.Stopped`
67. Напишите значение свойства `ThreadState`, которое сигнализирует о том, что поток приостановлен? – `ThreadState.Suspended`
68. Напишите значение свойства `ThreadState`, которое сигнализирует о том, что поток еще не был запущен на выполнение? – `ThreadState.Unstarted`
69. Напишите значение свойства `ThreadState`, которое сигнализирует о том, что для потока вызван метод `Abort ( )`, но остановка потока еще не произошла? – `ThreadState.AbortRequested`
70. Напишите значение свойства `ThreadState`, которое сигнализирует о том, что поток выполняется? – `ThreadState.Running`
71. Напишите значение свойства `ThreadState`, которое сигнализирует о том, что поток получил запрос на остановку? – `ThreadState.StopRequest`
72. Напишите значение свойства `ThreadState`, которое сигнализирует о том, что поток получил запрос на приостановку? – `ThreadState.SuspendRequest`
73. Напишите значение свойства `ThreadState`, которое сигнализирует о том, что поток заблокирован с помощью вызова методов `Sleep ( )` или `Join ( )`? – `ThreadState.WaitSleepJoin`
74. Какое количество состояний может принимать поток? – 10
75. Какое свойство позволяет получить информацию о потоке, который это свойство вызывает? – `CurrentThread`

## 5.2. Описание показателей и критериев контроля успеваемости, описание шкал оценивания

Для оценки знаний, умений, навыков и формирования компетенции по дисциплине применяется балльно-рейтинговая система контроля и оценки успеваемости обучающихся очной формы. Основные требования балльно-рейтинговой системы по дисциплине и шкала оценивания приведены в таблицах 5 и 7.

Таблица 5 – Требования балльно-рейтинговой системы по дисциплине

Виды работ	Количество видов работ	Максимальные баллы за подвид работы	Сроки выполнения подвидов работы	Дополнительные баллы	Штрафные баллы
					За нарушение сроков
Контрольные работы	2	по 2 баллов за работу	Каждые 2 недели	До +2 за 1 работу	До -2 за 1 работу
Выполнение домашних заданий	14	По 2 балла за 1 работу	еженедельно	До +1 балла за 1 работу	До -1 балла за 1 работу
Выполнение лабораторных работ	6	По 6 балла за 1 работу		До +4 баллов за 1 работу	
Посещение занятий (участие в обсуждениях задач)	11	До 2 балла за 1 неделю	еженедельно	Ответ у доски до +1 балла	По -1 баллу за 1 пропуск
Ответ на зачете с оценкой	1	10	январь		

Таблица 6 – Критерии оценивания результата обучения по дисциплине и шкала оценивания

Код и наименование компетенции	Код и наименование индикатора достижения компетенции	Критерии оценивания результатов обучения			
		Оценка «неудовлетворительно» / «не зачтено» 0-54% от тах рейтинговой оценки контроля	Оценка «удовлетворительно» / «зачтено» 55-70% от тах рейтинговой оценки контроля	Оценка «хорошо» / «зачтено» 71-85% от тах рейтинговой оценки контроля	Оценка «отлично» / «зачтено» 86-100% от тах рейтинговой оценки контроля
<b>ПСК-1</b> способен проектировать и разрабатывать компоненты программного обеспечения на основе современных парадигм, технологий и языков программирования	<b>ИПКС-1.1.</b> Использует методы и инструменты современных объектно-ориентированных языков программирования для разработки компонентов программного обеспечения.	Изложение учебного материала бессистемное, неполное, не знает основ многопоточного программирования, не может использовать методы многопоточного программирования в рамках поставленных целей и задач, что препятствует усвоению последующего материала	Фрагментарные, поверхностные знания по основам многопоточного программирования. Изложение полученных знаний неполное, однако это не препятствует усвоению последующего материала. Допускаются отдельные существенные ошибки, исправленные с помощью преподавателя. Затруднения при формулировании основных положений и их применении	Знает материал на достаточно хорошем уровне; представляет основные задачи в рамках постановки целей и выбора оптимальных способов их достижения.	Имеет глубокие знания всего материала, понимает структуру дисциплины; освоил новации лекционного курса по сравнению с учебной литературой; изложение полученных знаний полное, системное; допускаются единичные ошибки, самостоятельно исправляемые при собеседовании

Таблица 7 – Критерии оценивания

Оценка	Критерии оценивания
Высокий уровень «5» (отлично)	оценку « <b>отлично</b> » заслуживает обучающийся, освоивший знания, умения, компетенции и теоретический материал без пробелов; выполнивший все задания, предусмотренные учебным планом на высоком качественном уровне; практические навыки профессионального применения освоенных знаний сформированы.
Средний уровень «4» (хорошо)	оценку « <b>хорошо</b> » заслуживает обучающийся, практически полностью освоивший знания, умения, компетенции и теоретический материал, учебные задания не оценены максимальным числом баллов, в основном сформировал практические навыки.
Пороговый уровень «3» (удовлетворительно)	оценку « <b>удовлетворительно</b> » заслуживает обучающийся, частично с пробелами освоивший знания, умения, компетенции и теоретический материал, многие учебные задания либо не выполнил, либо они оценены числом баллов близким к минимальному, некоторые практические навыки не сформированы.
Минимальный уровень «2» (неудовлетворительно)	оценку « <b>неудовлетворительно</b> » заслуживает обучающийся, не освоивший знания, умения, компетенции и теоретический материал, учебные задания не выполнил, практические навыки не сформированы.

## 6. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

### 6.1. Учебная литература

**6.1.1 Биллиг, В.А.** Параллельные вычисления и многопоточное программирование: учебное пособие / В.А. Биллиг. — 2-е изд. — Москва : ИНТУИТ, 2016. — 310 с. — Текст: электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/100361>

**6.1.2 Иртегов, Д.В.** Многопоточное программирование с использованием POSIX Threads: учебное пособие / Д.В. Иртегов. — 2-е изд. — Москва : ИНТУИТ, 2016. — 170 с. — Текст : электронный // Лань: электронно-библиотечная система. — URL: <https://e.lanbook.com/book/100653>

**6.1.3 Водяхо, А.И.** Архитектурные решения информационных систем: учебник / А.И. Водяхо, Л.С. Выговский, В.А. Дубенецкий, В.В. Цехановский. — 2-е изд., перераб. — Санкт-Петербург: Лань, 2021. — 356 с. — ISBN 978-5-8114-2556-3. — Текст: электронный // Лань: электронно-библиотечная система. — URL: <https://e.lanbook.com/book/167464>

**6.1.4 Сердюк, Ю.П.** Параллельное программирование для многоядерных процессоров: учебное пособие / Ю.П. Сердюк, А. В. Петров. — 2-е изд. — Москва: ИНТУИТ, 2016. — 170 с. — Текст: электронный // Лань: электронно-библиотечная система. — URL: <https://e.lanbook.com/book/100357>

### 6.2. Методические указания, рекомендации и другие материалы к занятиям

**6.2.1** Объектно-ориентированное программирование: Методические указания к выполнению лабораторных работ по дисциплине «Технология программирования» / Сост. Н.О. Кулигина – Дзержинск, 2015

## 7. ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Учебный процесс по дисциплине обеспечен необходимым комплектом лицензионного и свободно распространяемого программного обеспечения, в том числе

отечественного производства (состав по дисциплине определен в настоящей РПД и подлежит обновлению при необходимости).

### 7.1. Перечень информационных справочных систем

Дисциплина, относится к группе дисциплин, в рамках которых предполагается использование информационных технологий как вспомогательного инструмента.

Информационные технологии применяются в следующих направлениях: при подготовке и оформлении отчетов о лабораторных работах, выполнении заданий для самостоятельной работы.

Таблица 8 – Перечень электронных библиотечных систем

№	Наименование ЭБС	Ссылка к ЭБС
1	Консультант студента	<a href="http://www.studentlibrary.ru/">http://www.studentlibrary.ru/</a>
2	Лань	<a href="https://e.lanbook.com/">https://e.lanbook.com/</a>
3	Виртуальная книжная полка НТБ НГТУ	<a href="http://cdot-nttu.ru/электронная_библиотека">http://cdot-nttu.ru/электронная_библиотека</a>
4	Информационная система «Единое окно доступа к образовательным ресурсам»	<a href="http://window.edu.ru/">http://window.edu.ru/</a>

### 7.2. Перечень лицензионного и свободно распространяемого программного обеспечения, в том числе отечественного производства, необходимого для освоения дисциплины

Таблица 9 – Программное обеспечение

№ п/п	Программное обеспечение, используемое в университете на договорной основе	Программное обеспечение свободного распространения
1	Microsoft Windows 10 (подписка MSDN 700593597, подписка DreamSparkPremium, 19.06.19)	Adobe Acrobat Reader <a href="https://acrobat.adobe.com/ru/ru/acrobat/pdf-reader.html">https://acrobat.adobe.com/ru/ru/acrobat/pdf-reader.html</a>
2	Microsoft office 2010 (Лицензия № 49487295 от 19.12.2011)	OpenOffice <a href="https://www.openoffice.org/ru/">https://www.openoffice.org/ru/</a>
3	Консультант Плюс	PTC Mathcad Express <a href="https://www.mathcad.com/ru">https://www.mathcad.com/ru</a>

### Перечень современных профессиональных баз данных и информационных справочных систем

В таблице 10 указан перечень профессиональных баз данных и информационных справочных систем, к которым обеспечен доступ (удаленный доступ).

Таблица 10 – Перечень современных профессиональных баз данных и информационных справочных систем

№ п/п	Наименование профессиональной базы данных, информационно-справочной системы	Доступ к ресурсу (удаленный доступ с указанием ссылки/доступ из локальной сети университета)
1	2	3
1	База данных стандартов и регламентов РОССТАНДАРТ	<a href="https://www.gost.ru/portal/gost//home/standarts">https://www.gost.ru/portal/gost//home/standarts</a>
2	Перечень профессиональных баз данных и информационных справочных систем	<a href="https://cyberpedia.su/21x47c0.html">https://cyberpedia.su/21x47c0.html</a>

3	Инструменты и веб-ресурсы для веб-разработки – 100+	<a href="https://techblog.sdstudio.top/blog/instrumenty-i-veb-resursy-dlia-veb-razrabotki-100-plus">https://techblog.sdstudio.top/blog/instrumenty-i-veb-resursy-dlia-veb-razrabotki-100-plus</a>
4	Справочная правовая система «КонсультантПлюс»	доступ из локальной сети

## 8. ОБРАЗОВАТЕЛЬНЫЕ РЕСУРСЫ ДЛЯ ИНВАЛИДОВ И ЛИЦ С ОВЗ

В таблице 11 указан перечень образовательных ресурсов, имеющих формы, адаптированные к ограничениям их здоровья, а также сведения о наличии специальных технических средств обучения коллективного и индивидуального пользования.

Таблица 11 – Образовательные ресурсы для инвалидов и лиц с ОВЗ

№	Перечень образовательных ресурсов, приспособленных для использования инвалидами и лицами с ОВЗ	Сведения о наличии специальных технических средств обучения коллективного и индивидуального пользования
1	ЭБС «Консультант студента»	озвучка книг и увеличение шрифта
2	ЭБС «Лань»	специальное мобильное приложение – синтезатор речи, который воспроизводит тексты книг и меню навигации
3	ЭБС «Юрайт»	версия для слабовидящих

Согласно Федеральному Закону об образовании 273-ФЗ от 29.12.2012 г. ст. 79, п.8 "Профессиональное обучение и профессиональное образование обучающихся с ограниченными возможностями здоровья осуществляются на основе образовательных программ, адаптированных при необходимости для обучения указанных обучающихся". АОП разрабатывается по каждой направленности при наличии заявлений от обучающихся, являющихся инвалидами или лицами с ОВЗ и изъявивших желание об обучении по данному типу образовательных программ.

## 9. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ, НЕОБХОДИМОЕ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ

Учебные аудитории для проведения занятий по дисциплине, оснащены оборудованием и техническими средствами обучения.

В таблице 12 перечислены:

- учебные аудитории для проведения учебных занятий, оснащенные оборудованием и техническими средствами обучения;
- помещения для самостоятельной работы обучающихся, которые оснащены компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду ДПИ НГТУ.

Таблица 12 – Оснащенность аудиторий и помещений для самостоятельной работы обучающихся по дисциплине

№	Наименование аудиторий и помещений для самостоятельной работы	Оснащенность аудиторий помещений и помещений для самостоятельной работы	Перечень лицензионного программного обеспечения. Реквизиты подтверждающего документа
---	---	---	---

1	<b>1433А</b> Аудитория для лекционных и практических занятий Нижегородская обл., г. Дзержинск, ул. Гайдара, д. 49	Комплект демонстрационного оборудования: ПК, с выходом на мультимедийный проектор, на базе Intel Pentium G4560 3.5 ГГц, 4 Гб ОЗУ, монитор 20' – 1шт. Мультимедийный проектор Epson- 1 шт; Экран – 1 шт.	<ul style="list-style-type: none"> <li>• Microsoft Windows 10 Домашняя (поставка с ПК)</li> <li>• LibreOffice 6.1.2.1. (свободное ПО)</li> <li>• Foxit Reader (свободное ПО);</li> <li>• 7-zip для Windows (свободное ПО)</li> </ul>
2	<b>1234</b> Научно-техническая библиотека ДПИ НГТУ, студенческий читальный зал; Нижегородская обл., г. Дзержинск, ул. Гайдара, д. 49	Комплект демонстрационного оборудования: ПК, с выходом на мультимедийный проектор, на базе Intel Pentium G4560 3.5 ГГц, 4 Гб ОЗУ, монитор 20' – 1шт. Мультимедийный проектор Epson- 1 шт; Экран – 1 шт.; Набор учебно-наглядных пособий	<ul style="list-style-type: none"> <li>• Microsoft Windows 10 Домашняя (поставка с ПК)</li> <li>• LibreOffice 6.1.2.1. (свободное ПО)</li> <li>• Foxit Reader (свободное ПО);</li> <li>• 7-zip для Windows (свободное ПО)</li> </ul>
3	<b>1443а</b> компьютерный класс - помещение для СРС, курсового проектирования (выполнения курсовых работ), Нижегородская обл., г. Дзержинск, ул. Гайдара, д. 49	<ul style="list-style-type: none"> <li>• ПК на базе Intel Celeron 2.67 ГГц, 2 Гб ОЗУ, монитор Acer 17' – 4 шт.</li> </ul> ПК подключены к сети «Интернет» и обеспечивают доступ в электронную информационно-образовательную среду университета	<ul style="list-style-type: none"> <li>• Microsoft Windows 7 (подписка DreamSpark Premium)</li> <li>• Apache OpenOffice 4.1.8 (свободное ПО);</li> <li>• Mozilla Firefox (свободное ПО);</li> <li>• Adobe Acrobat Reader (свободное ПО);</li> <li>• 7-zip для Windows (свободное ПО);</li> <li>• КонсультантПлюс (ГПД № 0332100025418000079 от 21.12.2018);</li> </ul>

## 10. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ОБУЧАЮЩИМСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

### 10.1. Общие методические рекомендации для обучающихся по освоению дисциплины, образовательные технологии

Дисциплина реализуется посредством проведения контактной работы с обучающимися (включая проведение текущего контроля успеваемости), самостоятельной работы обучающихся и промежуточной аттестации.

Контактная работа: аудиторная, внеаудиторная, а также проводится в электронной информационно-образовательной среде университета (далее - ЭИОС).

Преподавание дисциплины ведется с применением следующих видов образовательных технологий:

- балльно-рейтинговая технология оценивания;

При преподавании дисциплины «Многопоточное программирование», используются современные образовательные технологии, позволяющие повысить активность обучающихся при освоении материала курса и предоставить им возможность эффективно реализовать часы самостоятельной работы.

Весь лекционный материал курса сопровождается компьютерными презентациями, в которых наглядно преподносится материал различных разделов курса, что дает возможность обсудить материал с обучающимися во время чтения лекций, активировать

их деятельность при освоении материала. Материалы лекций в виде слайдов находятся в свободном доступе на в системе MOODLE и могут быть получены до чтения лекций и проработаны обучающимися в ходе самостоятельной работы.

На лекциях и практических занятиях реализуются интерактивные технологии, приветствуются вопросы и обсуждения, используется личностно-ориентированный подход, технология работы в малых группах, что позволяет обучающимся проявить себя, получить навыки самостоятельного изучения материала, выровнять уровень знаний в группе.

Все вопросы, возникшие при самостоятельной работе над домашним заданием, подробно разбираются на практических занятиях и лекциях. Проводятся индивидуальные и групповые консультации с использованием как встреч с обучающимися, так и современных информационных технологий (электронная почта, Zoom).

Иницируется активность обучающихся, поощряется задание любых вопросов по материалу, практикуется индивидуальный ответ на вопросы обучающегося, рекомендуются методы успешного самостоятельного усвоения материала в зависимости от уровня его базовой подготовки.

Для оценки знаний, умений, навыков и уровня сформированности компетенции применяется балльно-рейтинговая система контроля и оценки успеваемости обучающихся в процессе текущего контроля.

Промежуточная аттестация проводится в форме зачета с оценкой с учетом текущей успеваемости.

**Результат обучения считается сформированным на повышенном уровне**, если теоретическое содержание курса освоено полностью. При устных собеседованиях обучающийся исчерпывающе, последовательно, четко и логически излагает учебный материал; свободно справляется с задачами, вопросами и другими видами заданий, использует в ответе дополнительный материал. Все предусмотренные рабочей учебной программой задания выполнены в соответствии с установленными требованиями, обучающийся способен анализировать полученные результаты, проявляет самостоятельность при выполнении заданий.

**Результат обучения считается сформированным на пороговом уровне**, если теоретическое содержание курса в основном освоено. При устных собеседованиях обучающийся последовательно излагает учебный материал; при затруднениях способен после наводящих вопросов продолжить обсуждение, справляется с задачами, вопросами и другими видами заданий, требующих применения знаний; все предусмотренные рабочей учебной программой задания выполнены в соответствии с установленными требованиями, обучающийся способен анализировать полученные результаты; проявляет самостоятельность при выполнении заданий

**Результат обучения считается несформированным**, если обучающийся при выполнении заданий не демонстрирует знаний учебного материала, допускает ошибки, неуверенно, с большими затруднениями выполняет задания, не демонстрирует необходимых умений, качество выполненных заданий не соответствует установленным требованиям, качество их выполнения оценено числом баллов ниже трех по оценочной системе, что соответствует допороговому уровню.

## **10.2. Методические указания для занятий лекционного типа**

Лекционный курс предполагает систематизированное изложение основных вопросов тематического плана. В ходе лекционных занятий раскрываются базовые вопросы в рамках каждой темы дисциплины (таблица 4). Обозначаются ключевые аспекты тем, а также делаются акценты на наиболее сложные и важные положения изучаемого материала. Материалы лекций являются опорной основой для подготовки обучающихся к практическим занятиям и выполнения заданий самостоятельной работы, а также к



мероприятиям текущего контроля успеваемости и промежуточной аттестации по дисциплине.

### **10.3. Методические указания по освоению дисциплины на лабораторных работах**

Подготовку к каждой лабораторной работе обучающийся должен начать с ознакомления с планом занятия, который отражает содержание предложенной темы. Каждая выполненная работа с оформленным отчетом и подлежит защите у преподавателя.

При оценивании лабораторных работ учитывается следующее:

- качество выполнения части работы, посвященной программной реализации алгоритма, и степень соответствия результатов работы заданным требованиям;
- качество оформления отчета по работе;
- качество устных ответов на контрольные вопросы при защите работы.

### **10.4. Методические указания по самостоятельной работе обучающихся**

Самостоятельная работа обеспечивает подготовку обучающихся к аудиторным занятиям и мероприятиям текущего контроля и промежуточной аттестации по изучаемой дисциплине. Результаты этой подготовки проявляются в активности обучающихся на занятиях и в качестве выполненных заданий для самостоятельной работы и других форм текущего контроля.

При выполнении заданий для самостоятельной работы рекомендуется проработка материалов лекций по каждой пройденной теме, а также изучение рекомендуемой литературы, представленной в Разделе 6.

В процессе самостоятельной работы при изучении дисциплины обучающиеся могут работать на компьютере в специализированных аудиториях для самостоятельной работы (таблица 12). В аудиториях имеется доступ через информационно-телекоммуникационную сеть «Интернет» к электронной информационно-образовательной среде университета (ЭИОС) и электронной библиотечной системе (ЭБС), где в электронном виде располагаются учебные и учебно-методические материалы, которые могут быть использованы для самостоятельной работы при изучении дисциплины.

## **11. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ КОНТРОЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ**

### **11.1. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта в ходе текущего контроля успеваемости**

Для текущего контроля знаний обучающихся по дисциплине проводится **комплексная оценка знаний**, включающая

- проведение лабораторных работ;
- проведение контрольных работ для обучающихся очной формы;
- выполнение заданий для самостоятельной работы для обучающихся очной формы;
- зачет с оценкой.

#### **11.1.1. Типовые задания для лабораторных работ**

##### **Задание № 1:**

Реализовать многопоточное приложение с помощью технологии Windows Forms и языка программирования C#, используя механизм синхронизации, реализованный в классе Mutex. Приложение должно содержать два синхронизированных между собой метода:

Первый метод – запись информации о новом студенте в файл по нажатию на кнопку/клавишу (общий ресурс для двух потоков).

Второй метод – вывод (чтение из файла) информации о студентах (фоновый поток).

Общий ресурс – текстовый файл, в котором построчно находится следующая информация: ФИО студента, группа (например, Иванов А. А., группа 17ПМ).

**Задание № 2:**

Реализовать многопоточное приложение с помощью технологии Windows Forms и языка программирования C#, используя механизм синхронизации, реализованный в классе Semaphore.

Реализовать приложение «онлайн-библиотеку» с ограниченным количеством читателей (не больше 5 человек): один читатель – один поток. Время пребывания читателя в библиотеке реализовать с помощью класса Random.

**Задание № 3:**

Реализовать многопоточное приложение с помощью технологии Windows Forms и языка программирования C#, используя класс Thread.

Реализовать приложение «текстовый редактор», которое подсчитывало бы количество слов и символов в тексте, который ввел пользователь.

Первый метод – подсчет количества слов в тексте (первый поток)

Второй метод – подсчет количества символов в тексте (второй поток)

**Задание № 4:**

Реализовать консольное многопоточное приложение, используя механизм синхронизации, реализованный в классе Monitor.

Реализовать приложение, содержащее один метод для подсчета суммы и среднего арифметического всех элементов массива, к которому одновременно обращаются несколько потоков.

Общий ресурс – статический массив

Каждый поток использует в качестве входного метода – метод для подсчета суммы и среднего арифметического всех элементов массива.

**Задание № 5:**

Реализовать консольное многопоточное приложение, используя механизм синхронизации, реализованный в классе Barrier.

Реализовать приложение, которое подсчитывает количество итераций (или этапов), необходимых для того, чтобы каждый из двух потоков одновременно нашел свое частичное решение: правильно составил 2 части одного предложения, чтобы получилась верная фраза, используя алгоритм перемешивания слов. В каждом потоке должны перемешиваться свои слова, а затем операция барьера после этапа должна сравнивать два результата и проверять, правильно ли собрано полное предложение.

**Задание № 6:**

Реализовать консольное многопоточное приложение, используя механизм синхронизации, реализованный в классе Interlocked.

Реализовать приложение, которое определяет количество случайных чисел в диапазоне от 0 до 1 000, необходимое для создания 1 000 случайных чисел, равных среднему значению диапазона. Для подсчета количества случайных чисел использовать метод Increment ( ). Подсчет должен производиться с помощью нескольких потоков, а количество случайных чисел записываться в одну и ту же переменную для всех потоков.

**Задание № 7:**

Реализовать консольное многопоточное приложение, используя механизм синхронизации, реализованный в классе ReaderWriterLockSlim.

Реализовать приложение, которое с помощью нескольких потоков обращалось бы к кэшу данных, внутри которого хранится некоторая информация. Реализовать методы записи, извлечения и замены значения кэша, используя функционал класса ReaderWriterLockSlim.

**Задание № 8:**

Реализовать с помощью класса Parallel и метода Parallel.ForEach приложение для изменения изображений. Реализовать приложение, которое бы выполняло разворот на 90 градусов изображений формата JPEG, которые находятся по определенному адресу на компьютере. После чего повернутые изображения должны быть сохранены в новую папку.

### **11.1.2. Типовые задания для самостоятельной работы обучающихся очной формы**

1. Какой метод структуры SpinLock снимает блокировку с объекта (или общего ресурса) для выполнения работы другого потока? - Exit ( )
2. Какой метод структуры SpinLock имеет возможность отслеживать возможность получить блокировку объект (или общий ресурс) на время выполнения работы потока? - TryEnter ( )
3. Укажите форму определения структуры SpinLock? – public struct SpinLock
4. С помощью какого свойства структуры SpinLock можно определить, имеет ли какой-либо поток блокировку в настоящий момент? – IsHeld
5. С помощью какого свойства структуры SpinLock можно определить, имеет ли текущий поток блокировку в настоящий момент? – IsHeldByCurrentThread
6. Укажите наиболее употребляемые формы определения класса Barrier? – public Barrier (int participantCount) и public Barrier (int participantCount, Action<System.Threading.Barrier> postPhaseAction)
7. Какие значения передаются в качестве аргумента при создании экземпляра класса Barrier? – Количество участвующих потоков и Действие, выполняемое после каждой фазы
8. С помощью какого свойства класса Barrier можно получить номер текущей фазы барьера? – CurrentPhaseNumber
9. С помощью какого свойства класса Barrier можно получить общее количество участников? – ParticipantCount
10. С помощью какого свойства класса Barrier можно получить общее количество участников, которые еще не создали сигнал в текущей фазе? – ParticipantsRemaining
11. Какой метод уведомляет Barrier о добавлении дополнительного участника? – AddParticipant ( )
12. Укажите форму определения метода AddParticipant класса Barrier? – public long AddParticipant ( )
13. Какой метод уведомляет Barrier о добавлении нескольких дополнительных участников? – AddParticipants ( )
14. Укажите форму определения метода AddParticipants класса Barrier? – public long AddParticipants (int participantCount)
15. Какой метод уведомляет Barrier об удалении одного участника? – RemoveParticipant
16. Укажите форму определения метода RemoveParticipant? – public void RemoveParticipant ( )
17. Какой метод уведомляет Barrier об удалении нескольких участников? – RemoveParticipants
18. Укажите форму определения метода RemoveParticipants? – public void RemoveParticipants (int participantCount)
19. С помощью какого метода участник может сообщить, что достиг барьера и ожидает достижения барьера другими участниками? – SignalAndWait
20. Укажите правильную форму объявления конструктора класса ManualResetEvent? – public ManualResetEvent (bool initialState)
21. Укажите правильную форму объявления конструктора класса AutoResetEvent? – public AutoResetEvent (bool initialState)
22. Какой аргумент необходимо передать в конструктор класса ManualResetEvent? – логическое значение true или false
23. Какой аргумент необходимо передать в конструктор класса AutoResetEvent? – логическое значение true или false
24. С помощью какого метода класса ManualResetEvent выполнение вызывающего потока приостанавливается до тех пор, пока не будет получено уведомление о событии? – WaitOne
25. С помощью какого метода класса ManualResetEvent поток уведомляется о наступлении некоторого события? – Set ( )
26. С помощью какого метода класса ManualResetEvent событийный объект возвращается в несигнальное состояние? – Reset ( )
27. Укажите наиболее употребляемую форму определения метода Set класса ManualResetEvent? – public bool Set ( )

28. Укажите наиболее употребляемую форму определения метода WaitOne класса ManualResetEvent? – public virtual bool WaitOne ()
29. Укажите наиболее употребляемую форму определения метода Reset класса ManualResetEvent? – public bool Reset ()
30. С помощью какого метода класса AutoResetEvent выполнение вызывающего потока приостанавливается до тех пор, пока не будет получено уведомление о событии? – WaitOne
31. С помощью какого метода класса AutoResetEvent поток уведомляется о наступлении некоторого события? – Set ()

**11.2. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта в ходе промежуточной аттестации по дисциплине**

1. Укажите наиболее употребляемую форму определения метода Set класса AutoResetEvent? – public bool Set ()
2. Укажите наиболее употребляемую форму определения метода WaitOne класса AutoResetEvent? – public virtual bool WaitOne ()
3. В чем принципиальное отличие использования класса ManualResetEvent от класса AutoResetEvent? – для события типа AutoResetEvent событийный объект автоматически переходит в несигнальное состояние, как только поток, ожидающий это событие, получит уведомление о нем и возобновит свое выполнение
4. Укажите наиболее употребляемые формы определения класса Mutex? – public Mutex() и public Mutex(bool initiallyOwned)
5. Какой аргумент передается в конструктор класса Mutex? – логическая переменная, указывающая на изначальную блокировку Мьютекса вызывающим потоком
6. С помощью какого метода класса Mutex блокируется доступ к общему ресурсу или объекту другим потокам? - WaitOne
7. Укажите наиболее употребляемую форму определения метода WaitOne класса Mutex? – public virtual bool WaitOne ();
8. С помощью какого метода класса Mutex освобождается ранее заблокированный объект? - ReleaseMutex
9. Укажите наиболее употребляемую форму определения метода ReleaseMutex класса Mutex? – public void ReleaseMutex ()
10. В чем выгодное отличие класса Semaphore от класса Mutex? – предоставляет одновременный доступ к общему ресурсу не одному, а нескольким потокам
11. Укажите наиболее употребляемую форму определения класса Semaphore? – public Semaphore(int initialCount, int maximumCount)
12. Какие аргументы передаются в конструктор класса Semaphore? - количество первоначально доступных разрешений и максимальное количество разрешений
13. С помощью какого метода класса Semaphore блокируется доступ к общему ресурсу или объекту другим потокам? – WaitOne
14. Укажите наиболее употребляемую форму определения метода WaitOne класса Semaphore? – public virtual bool WaitOne ();
15. С помощью какого метода класса Semaphore освобождается ранее заблокированный объект? – Release
16. Укажите наиболее употребляемую форму определения метода Release класса Semaphore? – public int Release () и public int Release (int releaseCount)
17. Какое значение необходимо передать в качестве аргумента в метод Release класса Semaphore? – количество разрешений для объекта Семафор
18. С помощью какого метода класса Monitor поток переходит в режим ожидания разблокировки объекта (или общего ресурса) на время выполнения работы другого потока? - Wait ()
19. Укажите наиболее употребляемые формы определения метода Wait? – public static void Wait (object waitObj) и public static void Wait (object waitObj, int milliseconds)

20. Что необходимо передать в качестве аргумента в метод `Wait ( )` класса `Monitor`? – переменную ссылочного типа, к которой ожидается доступ
21. С помощью каких методов класса `Monitor` возобновляется выполнение работы потока? – `Pulse ( )` и `PulseAll ( )`
22. Какой метод сообщает ожидающему потоку, стоящему первым в очереди потоков, о снятии блокировки с объекта? – `Pulse ( )`
23. Какой метод сообщает всем ожидающим потокам о снятии блокировки с объекта? – `PulseAll ( )`
24. Укажите наиболее употребляемую форму определения метода `Pulse ( )`? – `public static void Pulse (object waitObj)`
25. Укажите наиболее употребляемую форму определения метода `PulseAll ( )`? – `public static void Pulse (object waitObj)`
26. Что необходимо передать в качестве аргумента в метод `Pulse ( )` класса `Monitor`? – переменную ссылочного типа, освобождаемую от блокировки
27. Что необходимо передать в качестве аргумента в метод `PulseAll ( )` класса `Monitor`? – переменную ссылочного типа, освобождаемую от блокировки
28. Методы `Wait`, `Pulse` и `PulseAll` должны вызываться внутри какого блока кода? – `lock`
29. Какое исключение сгенерируется, если методы `Wait`, `Pulse` и `PulseAll` будут вызваны вне `lock`-блока кода? – `SynchronizationLockException`
30. С помощью какого метода можно отменить завершение работы потока, который сгенерировал исключение типа `ThreadAbortException`? – `ResetAbort`
31. Какой формат определения у метода `ResetAbort`? – `public void ResetAbort ( )`
32. Какое свойство позволяет определить состояние потока? – `ThreadState`
33. Напишите значение свойства `ThreadState`, которое сигнализирует об остановке потока, но не окончательном его завершении? – `ThreadState.Aborted`
34. Напишите значение свойства `ThreadState`, которое сигнализирует о том, что поток выполняется в фоновом режиме? – `ThreadState.Background`
35. Напишите значение свойства `ThreadState`, которое сигнализирует о том, что поток завершен? – `ThreadState.Stopped`
36. Напишите значение свойства `ThreadState`, которое сигнализирует о том, что поток приостановлен? – `ThreadState.Suspended`
37. Напишите значение свойства `ThreadState`, которое сигнализирует о том, что поток еще не был запущен на выполнение? – `ThreadState.Unstarted`
38. Напишите значение свойства `ThreadState`, которое сигнализирует о том, что для потока вызван метод `Abort ( )`, но остановка потока еще не произошла? – `ThreadState.AbortRequested`
39. Напишите значение свойства `ThreadState`, которое сигнализирует о том, что поток выполняется? – `ThreadState.Running`
40. Напишите значение свойства `ThreadState`, которое сигнализирует о том, что поток получил запрос на остановку? – `ThreadState.StopRequest`
41. Напишите значение свойства `ThreadState`, которое сигнализирует о том, что поток получил запрос на приостановку? – `ThreadState.SuspendRequest`
42. Напишите значение свойства `ThreadState`, которое сигнализирует о том, что поток заблокирован с помощью вызова методов `Sleep ( )` или `Join ( )`? – `ThreadState.WaitSleepJoin`
43. Какое количество состояний может принимать поток? – 10
44. Какое свойство позволяет получить информацию о потоке, который это свойство вызывает? – `CurrentThread`

Полный фонд оценочных средств для проведения промежуточной аттестации в форме компьютерного тестирования размещен в банке вопросов данного курса дисциплины в СДО MOODLE.