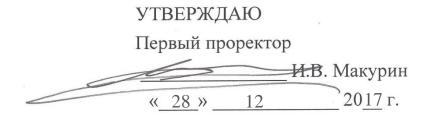
Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования «Комсомольский-на-Амуре государственный университет»

Кафедра «Математическое обеспечение и применение ЭВМ»



#### РАБОЧАЯ ПРОГРАММА

#### дисциплины «Параллельное программирование»

основной профессиональной образовательной программы подготовки бакалавров по направлению 09.03.01 — «Информатика и вычислительная техника» профиль «Программное обеспечение средств вычислительной техники и автоматизированных систем»

Форма обучения

заочная

Гехнология обучения

традиционная

Автор рабочей программы Профессор, доктор физикоматематических наук, профессор В.А.Тихомиров «14 » 04 2017 г.

#### СОГЛАСОВАНО

Директор библиотеки

*У* И.А. Романовская « 18 » 04 20<u>17</u> г.

Заведующий кафедрой «Математическое обеспечение и применение ЭВМ», кандидат технических наук, профессор

В.А. Тихомиров «16» 04 2017 г.

Заведующий выпускающей кафедрой «Математическое обеспечение и применение ЭВМ», кандидат технических наук, профессор

В.А. Тихомиров «16» 04 2017 г.

Декан ФЗДО, кандидат технических наук, доцент

Начальник учебно-методического управления

М.В.Семибратова
 «22 » 04 2017 г.
 — Е.Е. Поздеева
 «23 » 04 2017 г.

#### Введение

Рабочая программа дисциплины «Параллельное программирование» составлена в соответствии с требованиями федерального государственного образовательного стандарта высшего образования, утвержденного приказом Министерства образования и науки Российской Федерации от 12.01.2016 № 5, и основной профессиональной образовательной программы подготовки бакалавров по направлению 09.03.01 «Информатика и вычислительная техника». Данная рабочая программа подготовлена для студентов набора 2017 года и далее.

#### 1 Аннотация дисциплины

Наименование	Параллельное программирование									
дисциплины										
Цель	Формирование у студентов практических навыков разработки парал-									
дисциплины	лельных	лельных программ и многопоточных приложений с использования								
					dows и объе	ктно-ор	иентирова	нного		
	языка про	граммі	ирования (	C++.						
Задачи	1. Получе	ние пра	актически	х навыков	в создания па	араллель	ных проц	ессов		
дисциплины	и работы	с ними	•							
	2. Получе	ние пра	актически	х навыков	в программно	ой реали	зации мно	0Г0-		
	поточных	-								
	3. Получе	ние пра	актически	х навыков	в программи	рования	методов о	син-		
		хронизации параллельных процессов и потоков.								
		4. Получение практических навыков построения и использования сетей								
	Петри.									
Основные					ые системы.					
разделы	Разработк									
дисциплины	Классичес			-						
	Моделиро				слений.					
Общая	4 3.e. / 144	акаде	мических	часа						
трудоемкость			Аудиторі	ная нагруз	зка, ч	CPC,	Про-	Bce		
дисциплины				1.0		Ч	межу-	го за		
	Семестр	Пог	П	Поб	Курсовое		ТОЧ	ce-		
	Ссмсстр	Лек	Пр.	Лаб.	проектиро-		ная ат-	местр		
		ции	занятия	работы	вание		теста-	,ч		
							ция, ч			
	8	4	0	8	0	128	4	144		
	семестр									
	ИТОГО:	4	0	8	0	128	4	144		

## 2 Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами образовательной программы

Дисциплина «Параллельное программирование» нацелена на формирование компетенций, знаний, умений и навыков, указанных в таблице 1.

Таблица 1 – Компетенции, знания, умения, навыки

Наименование и шифр	Перечень форм	мируемых знаний, ум	ений, навыков,			
компетенции, в форми-	предусмотре	ных образовательной программой				
ровании которой	Перечень знаний	Перечень умений	Перечень навыков			
принимает участие	(с указанием	(с указанием	с ука-занием			
дисциплина	шифра)	шифра)	шифра)			
ПК-2 способностью раз-	36(ПК-2-7)	У6(ПК-2-7)	Н6(ПК-2-7)			
рабатывать компоненты	Принципы по-	Разрабатывать	Навыками парал-			
аппаратно-программных	строения архитек-	многопоточные	лельного про-			
комплексов и баз дан-	туры программ-	приложения	граммирования			
ных, используя совре-	ного обеспечения					
менные инструменталь-	(ПО) и виды ар-					
ные средства и техноло-	хитектуры про-					
гии программирования	граммного обес-					
	печения					

### 3 Место дисциплины (модуля) в структуре образовательной программы

Дисциплина (модуль) «Параллельное программирование» изучается на четвертом курсе в восьмом семестре.

Дисциплина является обязательной, входит в состав блока Б1 «Дисциплины (модули)» и относится к вариативной части.

Для освоения дисциплины необходимы знания, умения, навыки, сформированные на предыдущих этапах освоения компетенции ПК -2 «Способность разрабатывать компоненты аппаратно-программных комплексов и баз данных, используя современные инструментальные средства и технологии программирования», в процессе изучения дисциплин «Программирование», «Программирование на языке высокого уровня», «Программирование мобильных устройств», «Компоненты операционных систем», «Технологии разработки программного обеспечения», «ЭВМ и переферийные устройства», «Альтернативные операционные системы», «Компьютерная графика».

4 Объем дисциплины (модуля) в зачетных единицах с указанием количества академических часов, выделенных на контактную работу обучающихся с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающихся

Общая трудоемкость (объем) дисциплины составляет 4 зачетные единицы, 144 академических часов.

Распределение объема дисциплины (модуля) по видам учебных занятий представлено в таблице 2.

Таблица 2 – Объем дисциплины (модуля) по видам учебных занятий

	Всего академ	ических часов
Объем дисциплины	Очная форма обучения	Заочная (очно-заочная) форма обучения
Общая трудоемкость дисциплины		144
Контактная аудиторная работа обучающих- ся с преподавателем (по видам учебных за- нятий), всего В том числе:		16
занятия лекционного типа (лекции и иные учебные занятия, предусматривающие пре-имущественную передачу учебной информации педагогическими работниками)		4
занятия семинарского типа (семинары, практические занятия, практикумы, лабораторные работы, коллоквиумы и иные аналогичные занятия)		8
Самостоятельная работа обучающихся и контактная работа, включающая групповые консультации, индивидуальную работу обучающихся с преподавателями (в том числе индивидуальные консультации); взаимодействие в электронной информационно-образовательной среде вуза		128 (в том числе индивидуальные консультации 4 часа)
Промежуточная аттестация обучающихся		4

В соответствии с учебным планом в рамках дисциплины «Параллельное программирование» студент выполняет контрольную работу.

## 5 Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

Таблица 3 – Структура и содержание дисциплины (модуля) для очного обучения

таолица 3 — структура и содержание дисциплины (мод	ули) дли очног	o ooyacm	1/1		
	Компонент	Трудо- емкость	Форма прове-	руемые) ј	емые (контроли- результаты осво- ения
Наименование разделов, тем и содержание материала	учебного пла-	(в часах)		Компете	Знания, умения,
	на			Н	навыки
				ции	
Раздел Паралл	ельное программ	ирование			
Тема 1 Установочная лекция Системы параллельной обра-	Лекция		Традиционная	ПК-2	36(ПК-2-7)
ботки данных и архитектура процессоров					
Виды вычислительных систем. Системы параллельной обра-					
ботки данных. Конвейер и матрица. Классификация Флинна.					
Пути повышения производительности: законы Мура, Гроша,		2			
Амдала и гипотеза Минского. Архитектуры с сокращенным		_			
набором команд, со сверхдлинным командным словом. Век-					
торные архитектуры. Многопроцессорные синхронные вы-					
числительные системы. Конвейерные и матричные системы.					
Матричные системы. Систолические системы.	Полития		C	ПК-2	36(ПК-2-7)
Тема Параллельное выполнение процессов. Разработка мно-	Лекция		С использова-	11K-2	30(11K-2-7)
гопоточных программ.			нием активных методов обуче-		
Понятие процесса. Создание параллельных процессов и ра-			ния		
боты с ними в операционной системе Windows средствами		2	пил		
языка программирования С++. Использование объектов син-		2			
хронизации для обмена данными между процессами. Распа-					
раллеливание рекурсивных подпрограмм. Преобразование					
рекурсивных подпрограмм в многопоточные приложения. Многопоточный метод сдваивания.					
Тема 1 Программирование параллельных вычислитель-	Лабораторная		С использова-	ПК-2	36(ПК-2-7)
ных процессов	работа	2	нием активных	1111-2	H6(ПК-2-7)
Программная реализация создания и работы двух параллель-	Puooru		методов обуче-		
трограммим решизации создания и рассты двух парамены			тегодов обуте		

Наименование разделов, тем и содержание материала	Компонент учебного пла- на	Трудо- емкость (в часах)	Форма прове- дения		емые (контроли- результаты осво- ения Знания, умения, навыки
				ции	1100001111
ных процессов. Синхронизация работы параллельных процессов с помощью средств операционной системы Windows.			ния		
<b>Тема 2</b> Методы распараллеливания рекурсивных подпрограмм Программная реализация создания параллельных потоков и работы с ними. Разработка параллельных программ с помощью многопоточного метода сдваивания и многопоточного метода перебора с возвратом.	Лабораторная работа	3	С использованием активных методов обучения	ПК-2	36(ПК-2-7) У6(ПК-2-7) Н6(ПК-2-7)
<b>Тема 3</b> Построение и программирование волновых систем с помощью сетей Петри Моделирование волновых систем с помощью сетей Петри. Программирование волновых систем. Использование классаканала, построенного на основе решения задачи о производителе и потребителе.	Лабораторная работа	3	С использованием активных методов обучения	ПК-2	36(ПК-2-7) У6(ПК-2-7) Н6(ПК-2-7)
	Самостоятель- ная работа обучающихся	64	Подготовка к лабораторным занятиям	ПК-2	36(ПК-2-7)
	Самостоятельная работа обучающихся	4	Индивидуаль- ные консульта- ции	ПК-2	36(ПК-2-7)
	Самостоятельная работа обучающихся	22	Изучение теоретических разделов дисциплины	ПК-2	36(ПК-2-7)

Наименование разделов, тем и содержание материала	Компонент учебного пла-	Трудо-	Форма прове-	Планируемые (контролируемые) результаты освоения  Компете Знания, умения,		
	на	(в часах)	дения	н ции	навыки	
	Самостоятельная работа обучающихся	10	Выполнение, оформление и подготовка к защите лабораторных работ	ПК-2	36(ПК-2-7) Н6(ПК-2-7)	
<ul> <li>Тема Решения проблемы сериализации с помощью семафоров</li> <li>Разработка многопоточного приложения, в котором синхронизация работы потоков осуществляется с помощью семафоров.</li> </ul>	Самостоятельная работа обучающихся	22	Выполнение, оформление и подготовка к защите РГР			
	Текущий кон- троль	2	Тест	ПК-2	36(ПК-2-7)	
	Текущий кон- троль	4	Защиты лабораторных работ	ПК-2	36(ПК-2-7) Н6(ПК-2-7)	
Промежуточная аттестация по дисциплине (итоговая оценка)		4				
ИТОГО по дисциплине	Лекции	4	-	-	-	
	Лабораторные работы	8	-	-	-	
	Самостоятель- ная работа обучающихся	128	-	-	-	
ИТОГО: общая трудоемкость дисциплины (часов) 144 в том числе с использованием активных методов обучения 20 часов						

## 6 Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (модулю)

Самостоятельная работа обучающихся, осваивающих дисциплину «Параллельное программирование», состоит из следующих компонентов:

- изучение теоретических разделов дисциплины;
- подготовка к лабораторным занятиям;
- выполнение, оформление и подготовка к защите лабораторных работ;
- выполнение, оформление и подготовка к защите контрольной работы.

Для успешного выполнения всех разделов самостоятельной работы учащимся рекомендуется использовать следующее учебно-методическое обеспечение:

- 1 Хусаинов, А.А. Архитектура вычислительных систем: учеб. пособие / А.А. Хусаинов , Н.Н. Михайлова. Комсомольск-на-Амуре: ГОУ ВПО «КнАГТУ», 2007. 123с. // Собственные электронные ресурсы КнАГТУ: виртуальная библиотека института новых информационных технологий Режим доступа: http://www.initkms.ru/library/main?.
- 2 РД ФГБОУ ВО «КнАГТУ» 013-2016. Текстовые студенческие работы. Правила оформления. Введ. 2016-03-04. Комсомольск-на-Амуре: ФГБОУ ВО «КнАГТУ», 2016. 55 с.

Рекомендуемый график выполнения самостоятельной работы студента представлен в таблице 4.

Ниже приведены общие рекомендации по организации самостоятельной работы студентов.

В процессе обучения, работа студента над выполнением учебного плана складывается из двух составляющих: одна из них — это аудиторная работа в вузе по расписанию занятий, другая — внеаудиторная самостоятельная работа. Задания и материалы для самостоятельной работы выдаются во время учебных занятий по расписанию, на этих же занятиях преподаватель осуществляет контроль выполнения самостоятельной работы.

Для формирования у студентов практических навыков параллельного программирования, все задания к лабораторным работам и контрольной работы посвящены программной реализации параллельных алгоритмов и разработке, отладке и тестированию параллельных программ и многопоточных приложений. В рамках подготовки к лабораторным занятиям и изучения теоретических разделов дисциплины студенту необходимо осуществить поиск, хранение, обработку и анализ информации в сети Интернет и в технической литературе, как при изучении принципов построения архитектуры ПО, параллельных алгоритмов и методов распараллеливания, так и при самостоятельном проектировании, конструировании, отладке и тестировании параллельных программ.

При выполнении лабораторных работ и контрольной работы студенту необходимо использовать принципы построения архитектуры программного

обеспечения (ПО) и виды архитектуры ПО, разрабатывать многопоточные приложения, что позволяет сформировать навыки параллельного программирования.

При подготовке к защите лабораторных работ и контрольной работы студенту необходимо обратить внимание как на проработку теоретических вопросов по данной теме, так и на обоснование выбора средств языка программирования при разработке ПО, а также тестирование разработанного ПО при различных входных данных и параметрах.

При оформлении отчетов к лабораторным работам и контрольной работе студенту необходимо студенту необходимо осуществить поиск, хранение, обработку и анализ информации в сети Интернет и в технической литературе. Также при оформлении отчетов необходимо строго следовать РД ФГБОУ ВО «КнАГТУ» 013-2016. «Текстовые студенческие работы. Правила оформления».

После успешного выполнения и защиты контрольной работы на лабораторном занятии отчет по контрольной работе студенту необходимо разместить в его личном кабинете, расположенном на официальном сайте университета в информационно-телекоммуникационной сети «Интернет» по адресу https://student.knastu.ru.

Таблица 4 – Рекомендуемый график выполнения самостоятельной работы студентов при 17-недельном семестре

Вид самостоятельной							Чи	сло ч	асов в	недел	Ю							Итого
работы	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	по видам работ
Подготовка к лабора- торным занятиям	4	4	4	2	4	4	4	4	4	4	4	4	4	2	4	4	4	64
Индивидуальные кон- сультации				2									2					4
Изучение теоретических разделов дисциплины		1	3	2	2	2	2	2						2	2	2	2	22
Выполнение, оформление и подготовка к защите лабораторных работ								1	1	1	1	1	1	1	1	1	1	10
Выполнение, оформление и подготовка к защите РГР	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2	3	22
Тест																	2	2
Защиты лабораторных работ																2	2	4
ИТОГО в 8 семестре	5	6	8	7	7	7	7	8	6	6	6	6	8	7	9	11	14	128

### 7 Фонд оценочных средств для проведения текущего контроля и промежуточной аттестации обучающихся по дисциплине (модулю)

Таблица 5 – Паспорт фонда оценочных средств

Контролируемые разделы (темы) дисциплины	Код контроли- руемой компе- тенции (или ее части)	Наименование оценочного средства	Показатели оценки
Тема1- Параллельное вы- полнение процес- сов	ПК-2-7	Защита лабораторной работы 1 Тест	Применяет функции WinAPI для создания параллельных процессов и работы с ними. Умеет использовать объекты синхронизации для обмена данными между процессами.
Тема2-Методы распараллеливания рекурсивных программ	ПК-2-7	Защита лабораторной работы 2 Тест	Применяет функции WinAPI для создания параллельных потоков и работы с ними. Умеет разрабатывать многопоточные программы на языке C++.
Тема3- Сети Петри и их применение. Вол- новые системы	ПК-2-7	Защита лабораторной работы 3 Тест	Умеет использовать сети Петри и волновые системы для моделирования работы параллельных программ. Владеет навыками разработки многопоточных программ, применяя семафоры и события для синхронизации работы потоков.
Тема4-Семафоры и события	ПК-2-7	Защита РГР Тест	Умеет применять алгоритмы Деккера и Петерсона для решения задач синхронизации. Умеет применять семафоры для решения задачи сериализации. Умеет разрабатывать многопоточные приложения с использованием событий.

Промежуточная аттестация проводится в форме итоговой оценки. Методические материалы, определяющие процедуры оценивания зна-

ний, умений, навыков, характеризующих этапы формирования компетенций, представлены в виде технологической карты дисциплины (таблица 6).

Таблица 6 – Технологическая карта

1 403	<u> тица 6 – 1</u>	CAHOJIO		и карта
	Наиме-	~	Шка	
	нование	Сроки	ла	
		вы-	оце-	Критерии
	оценоч-	пол-	нива-	оценивания
	ного	нения	ва-	·
	средства		ния	
	1	1	111171	восьмой семестр
		Промеж	уточная	аттестация в форме итоговой оценки
1.	Тест	cec-	4	4 – правильные ответы на все вопроса, 3- правильные
		сия	балла	ответы на три вопроса, 2- правильные ответы на два вопроса, 1 – правильный ответ на один вопрос, 0 - все ответы неправильные
2.	Защита	cec-	16	16-лабораторная работа выполнена полностью, пра-
	лабора-	сия	бал-	вильно, своевременно, даны полные ответы на допол-
	торной	<b>V</b> 1131	ЛОВ	нительные вопросы во время защиты работы, студент
	работы		лов	показал отличные владения навыками параллельного
	1, 2			программирования, отчет выполнен аккуратно и в со-
	1, 4			ответствии с предъявляемыми требованиями, 10-
				лабораторная работа выполнена с замечаниями, сту-
				дент показал хорошее владение навыками параллель-
				ного программирования, не выдержаны сроки выпол-
				нения работы, даны неполные ответы на дополнитель-
				ные вопросы во время защиты работы, 5- студент вы-
				полнил работу с существенными неточностями, не со-
				блюдены сроки выполнения работы, студент показал
				удовлетворительное владение навыками параллельного
				программирования, 0- задание не выполнено.
3.	Тест	cec-	4	4 – правильные ответы на все вопроса, 3- правильные
3.	1001	сия	балла	ответы на три вопроса, 2- правильные ответы на два
		CHA	Ousisia	вопроса, 1 – правильный ответ на один вопрос, 0 - все
				ответы неправильные
1	Doversma	000	14	
4.	Защита	cec-		14-лабораторная работа выполнена полностью, пра-
	лабора-	сия	бал-	вильно, своевременно, даны полные ответы на допол-
	торной		ЛОВ	нительные вопросы во время защиты работы, студент
	работы 3			показал отличные владения навыками параллельного
				программирования, отчет выполнен аккуратно и в со-
				ответствии с предъявляемыми требованиями, 8-
				лабораторная работа выполнена с замечаниями, сту-
				дент показал хорошее владение навыками параллель-
				ного программирования, не выдержаны сроки выпол-
				нения работы, даны неполные ответы на дополнитель-
				ные вопросы во время защиты работы, 4- студент вы-
				полнил работу с существенными неточностями, не со-
				блюдены сроки выполнения работы, студент показал
				удовлетворительное владение навыками параллельного
_	Т		4	программирования, 0- задание не выполнено.
5.	Тест	cec-	4	4 – правильные ответы на все вопроса, 3- правильные
		сия	балла	ответы на три вопроса, 2- правильные ответы на два
				вопроса, 1 – правильный ответ на один вопрос, 0 - все
				ответы неправильные
6.	Защита	cec-	36	36- РГР выполнена полностью, правильно, своевре-
	РГР	сия	бал-	менно, даны полные ответы на дополнительные вопро-
		<b></b>		, Amili nothilite of beth in Actionini ending

			Шка	
	Наиме-	Сроки	ла	
	нование	_		<b>Г</b> ритории
	оценоч-	вы-	оце-	Критерии
	ного	пол-	нива-	оценивания
	средства	нения	ва-	
	1 / /		ния	
			ЛОВ	сы во время защиты работы, студент показал отличные умения разработки многопоточных приложений и владения навыками параллельного программирования, отчет выполнен аккуратно и в соответствии с предъявляемыми требованиями, 22 - РГР выполнена с замечаниями, студент показал хорошее умения разработки многопоточных приложений и владения навыками параллельного программирования, но не выдержаны сроки выполнения работы, даны неполные ответы на дополнительные вопросы во время защиты работы, 10- студент выполнил работу с существенными неточностями, не соблюдены сроки выполнения работы, студент показал удовлетворительное умения разработки многопоточных приложений и владения навыками параллельного программирования.
	T		4	ного программирования, 0- задание не выполнено.
7.	Тест	сес-	4 балла	4 – правильные ответы на все вопроса, 3- правильные ответы на три вопроса, 2- правильные ответы на два вопроса, 1 – правильный ответ на один вопрос, 0 - все ответы неправильные
8.	Защита лабора- торной работы 4	сессия	18 бал- лов	18-лабораторная работа выполнена полностью, правильно, своевременно, даны полные ответы на дополнительные вопросы во время защиты работы, студент показал отличные владения навыками параллельного программирования, отчет выполнен аккуратно и в соответствии с предъявляемыми требованиями, 11-лабораторная работа выполнена с замечаниями, студент показал хорошее владение навыками параллельного программирования, не выдержаны сроки выполнения работы, даны неполные ответы на дополнительные вопросы во время защиты работы, 6- студент выполнил работу с существенными неточностями, не соблюдены сроки выполнения работы, студент показал удовлетворительное владение навыками параллельного программирования, 0- задание не выполнено.
ИТС	рго:	-	100	-
			бал-	
			ЛОВ	

- **Критерии оценки результатов обучения по дисциплине:** 0 64 % от максимально возможной суммы баллов «неудовлетворительно» (недостаточный уровень для текущей аттестации по дисциплине);
- 65 74 % от максимально возможной суммы баллов «удовлетворительно» (пороговый (минимальный) уровень);
- 75 84 % от максимально возможной суммы баллов «хорошо» (средний уровень); 85 100 % от максимально возможной суммы баллов «отлично» (высокий (максимальный) уровень)

#### Задания для текущего контроля

## Лабораторная работа 1 «Программирование параллельных вычислительных процессов»

Задание. Разработать программу, запускающую параллельно работающие процессы. Первый процесс выводит графические объекты согласно варианту (прямоугольники) в левое окно, а второй выводит графические объекты согласно варианту (треугольники) в правое окно. Варианты заданий приведены в таблице 7.

**Пояснение.** Область вывода состоит из двух окон: верхнего и нижнего в нечетных вариантах или левого и правого в четных вариантах. Каждый процесс выводит в свое окно фигуры заданного вида, у которых случайно изменяются цвет, размеры и положение.

Например, в варианте 15, процессы выводят геометрические фигуры в верхнее и нижнее окна, поскольку 15 — нечетное число. Первый процесс выводит в верхнее окно закрашенные круги. Второй процесс выводит в нижнее окно отрезки, у которых изменяются координаты концов.

Таблица 7 – Варианты заданий

Левое окно Правое	Прямоуголь ник	Окружность	Отрезок	Закрашен- ный круг	Закрашенн ый прямо- угольник
Прямоуго- льник	ı	1	3	5	7
Окружно-	2	-	9	11	13
Отрезок	4	6	-	15	17
Закрашен- ный круг	8	10	12	-	19
Закрашен- ный прямо- угольник	14	16	18	20	-
Эллипс	21	22	23	24	25

**Пояснение.** Область вывода состоит из двух окон: верхнего и нижнего в нечетных вариантах или левого и правого в четных вариантах. Каждый процесс выводит в свое окно фигуры заданного вида, у которых случайно изменяются цвет, размеры и положение.

Например, в варианте 15, процессы выводят геометрические фигуры в верхнее и нижнее окна, поскольку 15 — нечетное число. Первый процесс выводит в верхнее окно закрашенные круги. Второй процесс выводит в нижнее окно отрезки, у которых изменяются координаты концов.

### Лабораторная работа 2 «Методы распараллеливания рекурсивных подпрограмм»

Задание . Разработать программу, решающую данную задачу с помощью рекурсивной функции. Преобразовать эту функцию и главную программу в многопоточную программу. Потоковые функции выполняются со случайной задержкой.

#### Варианты заданий

- 1. Вычисление значений многочлена  $sin(x) \approx \sum_{k=0}^{n} \frac{(-1)^k x^{2k+1}}{(2k+1)!}$  по схеме Горнера методом сдваивания.
- 2. Перебор последовательностей целых чисел  $0 \le x_1 < x_2 < ... < x_m < n$  при заданных m и n.
- 3. Вычисление значений многочлена  $\cos(x) \approx 1 + \sum_{k=0}^{n} \frac{(-1)^k x^{2k}}{(2k)!}$  по схеме Горнера методом сдваивания.
- 4. Перебор последовательностей целых чисел  $(x_1, x_2, ..., x_m)$ , находящихся в диапазоне  $0 \le x_i \le n$  и попарно не равных между собой.
- 5. Вычисление значений многочлена  $\ln(1+x) \approx \sum_{k=0}^{n} (-1)^k \frac{x^{k+1}}{k+1}$  по схеме Горнера методом сдваивания.
- 6. Перебор неубывающих последовательностей целых чисел  $0 \le x_1 \le x_2 \le ... \le x_m \le n$  при заданных m и n.
- 7. Вычисление значений многочлена  $ch(x) \approx \sum_{k=0}^{n} \frac{x^{2k}}{(2k)!}$  по схеме Горнера методом сдваивания.
  - 8. Перебор расстановок восьми ферзей, не угрожающих друг другу.
- 9. Вычисление значений многочлена  $sh(x) \approx \sum_{k=0}^{n} \frac{x^{2k+1}}{(2k+1)!}$  по схеме Горнера методом сдваивания.

- 10. Перебор последовательностей, состоящих из букв A, B и C, не содержащих смежных идентичных подтекстов и имеющих заданную длину n
- 11. Вычисление значений многочлена  $\sum_{k=1}^{n} \frac{(-1)^{k-1} x^k}{2k-1}$  по схеме Горнера методом сдваивания.
  - 12. Перебор разменов монетами заданной суммы денег.
- 13. Вычисление значений  $x_n$ , заданных с помощью рекуррентных соотношений  $x_0 = a$ ,  $x_n = nx_{n-1} + (n+1)^2$ , методом сдваивания.
  - 14. Обменная поразрядная сортировка.
- 15. Вычисление значений  $x_n$ , заданных с помощью рекуррентных соотношений  $x_0 = a$ ,  $x_n = n^2 x_{n-1} + n + 1$ , методом сдваивания.
- 16. Перебор всех последовательностей целых чисел  $(x_0, x_1, ..., x_{n-1})$ , находящихся в диапазоне  $0 \le x_i < n$ .
- 17. Вычисление значений  $x_n$ , заданных с помощью рекуррентных соотношений  $x_0 = a$ ,  $x_n = \frac{x_{n-1}}{n} + \frac{1}{n+1}$ , методом сдваивания.
  - 18. Закраска вершин графа.
- 19. Вычисление значений  $x_n$ , заданных с помощью рекуррентных соотношений  $x_0 = a$ ,  $x_n = \frac{x_{n-1}}{n^2} + \frac{1}{n}$ , методом сдваивания.
  - 20. Перебор всех подмножеств конечного множества.
- 21. Вычисление значений  $x_n$ , заданных с помощью рекуррентных соотношений  $x_0 = a$ ,  $x_n = \frac{x_{n-1}}{n^3} + \frac{1}{n}$ , методом сдваивания.
- 22. Перебор всех последовательностей целых чисел  $0 < x_1 \le x_2 \le ... \le x_m$ , для которых  $x_1 + x_2 + ... + x_m = n$ . Числа m и n заданы.
- 23. Вычисление значений  $x_n$ , заданных с помощью рекуррентных соотношений  $x_0 = a$ ,  $x_n = \frac{x_{n-1}}{(n^2+1)} + n$ , методом сдваивания.
- 24. Перебор всех последовательностей положительных чисел  $(x_1,x_2,...,x_{2n})$ , удовлетворяющих неравенствам  $x_1 < x_2 > x_3 < x_4 > ... < x_{2n}$  и равенствам  $x_1 + x_2 + ... + x_{2n} = m$ . Числа n и m заданы.
- 25. Вычисление значений  $x_n$ , заданных с помощью рекуррентных соотношений  $x_0 = a$ ,  $x_n = \frac{x_{n-1}}{n^2 n + 1} + n$ , методом сдваивания.

#### РΓР

#### «Решения проблемы сериализации с помощью семафоров»

Задание. Разработать многопоточную программу для вычисления заданного логического выражения, состоящего из поразрядных операций, арифметического выражения или приближенного значения интеграла. Недостающие данные можно вводить с помощью генератора случайных чисел.

Программа должна запускать n потоков. Параллельные вычисления в потоках производятся с небольшими случайными задержками.

Проверить результат вычисления с помощью подпрограммы, состоящей из одного цикла.

#### Варианты заданий

- 1.  $(a_0 + b_0)(a_1 + b_1)...(a_{n-1} + b_{n-1})$ .
- 2.  $a_0 \& b_0 \lor a_1 \& b_1 \lor ... \lor a_{n-1} \& b_{n-1}$ .
- 3.  $\sin(a_0) + \sin(a_1) + ... + \sin(a_{n-1})$ .

4. 
$$\int_{0}^{\pi} \sin(x) dx \approx \frac{\pi}{n} \sum_{k=0}^{n-1} \sin\left(\frac{\pi k}{n}\right).$$

- 5.  $\min(a_0 + b_0, a_1 + b_1, ..., a_{n-1} + b_{n-1})$ .
- 6.  $a_0b_0 + a_1b_1 + ... + a_{n-1}b_{n-1}$ .
- 7.  $(a_0 \lor b_0) \& (a_1 \lor b_1) \& \dots \& (a_{n-1} \lor b_{n-1})$ .
- 8.  $arctg(a_0) + arctg(a_1) + ... + arctg(a_{n-1})$ .

9. 
$$\int_{0}^{a} arctg(x)dx \approx \frac{a}{n} \sum_{k=0}^{n-1} arctg\left(\frac{ka}{n}\right).$$

- 10.  $\max(a_0 b_0, a_1 b_1, ..., a_{n-1} b_{n-1})$ .
- 11.  $(a_0-b_0)(a_1-b_1)...(a_{n-1}-b_{n-1})$ .
- 12.  $(\neg a_0 \lor b_0) \& (\neg a_1 \lor b_1) \& \dots \& (\neg a_{n-1} \lor b_{n-1})$ .
- 13.  $\cos(a_0) + \cos(a_1) + \dots + \cos(a_{n-1})$ .

14. 
$$\int_{0}^{a} e^{-x^{2}} dx \approx \frac{a}{n} \sum_{k=0}^{n-1} exp\left(-\left(\frac{ka}{n}\right)^{2}\right).$$

- 15.  $\max(a_0, b_0) + \max(a_1, b_1) + ... + \max(a_{n-1}, b_{n-1})$ .
- 16.  $\sum_{k=0}^{n-1} (-1)^k a_k b_k$
- 17.  $\max(\min(a_0,b_0),\min(a_1,b_1),...,\min(a_{n-1},b_{n-1}))$ .
- 18.  $\exp(-a_0^2) + \exp(-a_1^2) + ... + \exp(-a_{n-1}^2)$ .
- 19.  $\int_{0}^{a} \cos(x) dx \approx \frac{a}{n} \sum_{k=0}^{n-1} \cos\left(\frac{ak}{n}\right).$
- 20.  $\min(a_0b_0, a_1b_1, ..., a_{n-1}b_{n-1})$ .

21. 
$$\sum_{k=0}^{n-1} (-1)^k (a_k + b_k).$$

22. 
$$e^{-x} \approx \sum_{k=0}^{n-1} (-1)^k \frac{x^k}{k!}$$
, где  $-1 < x < 1$ .

23.  $\max(\sin(a_0), \sin(a_1), ..., \sin(a_{n-1}))$ .

24. 
$$\int_{0}^{a} \frac{dx}{1+x^{4}} \approx \frac{a}{n} \sum_{k=0}^{n-1} \left( 1 / \left( 1 + \left( \frac{ak}{n} \right)^{4} \right) \right).$$

25. 
$$\xi(x) \approx \sum_{k=1}^{n} \frac{1}{k^{s}}$$
, где  $1 < s < \infty$ .

# Лабораторная работа 3 «Построение и программирование волновых систем с помощью сетей Петри»

 $y_n = \exp(u_n) * \cos(v_n)$ .

 $y_n = \exp(\sin(u_n) + \sin(v_n))$ .

 $y_n = u_n * (\sin(u_n) + \cos(v_n)).$ 

 $y_n = \cos(u_n - v_n).$ 

Задание. Разработать программу, которая, согласно варианту, загружает потоки, реализующие волновую систему. Привести сеть Петри этой волновой системы.

#### Варианты заданий

1.  $x_n = \sin(u_n + v_n),$ 

14.  $x_n = \sin(\cos(u_n))$ ,

16.  $x_n = \sin(\exp(u_n))$ ,

 $15. x_n = \sin(u_n) * \exp(v_n),$ 

$2.  x_n = \cos(u_n - v_n),$	$y_n = \sin(u_n) * \exp(v_n).$
3. $x_n = u_n * (\sin(u_n) + \cos(v_n)),$	$y_n = \sin(\exp(v_n)).$
$4.  x_n = \exp(\sin(u_n)),$	$y_n = \sin(u_n * v_n).$
$5.  x_n = \cos(u_n * v_n) ,$	$y_n = \cos(\exp(v_n) + u_n).$
$6.  x_n = \sin(u_n * v_n) ,$	$y_n = \exp(\sin(u_n + v_n)).$
$7.  x_n = \exp(u_n) * \cos(v_n) ,$	$y_n = \cos(u_n - v_n).$
8. $x_n = \sin(u_n) * \exp(v_n),$	$y_n = \sin(\cos(v_n)).$
$9.  x_n = \cos(\sin(u_n + v_n)),$	$y_n = \sin(u_n + v_n).$
$10. x_n = u_n * v_n + \sin(v_n),$	$y_n = \cos(\sin(\cos(v_n))).$
$11. x_n = \exp(\cos(u_n)),$	$y_n = (\sin(u_n) + \cos(v_n)) * \exp(v_n).$
$12. x_n = \sin(u_n * v_n),$	$y_n = \exp(\sin(u_n + v_n)).$
$13. x_n = \cos(u_n - v_n),$	$y_n = \exp(\cos(u_n + v_n)).$

```
17. x_n = \cos(u_n * v_n),
                                             y_n = \exp(\cos(v_n)).
18. x_n = \cos(\exp(u_n) + v_n),
                                             y_n = \sin(u_n * v_n).
19. x_n = \exp(\cos(u_n + v_n)),
                                             y_n = \sin(u_n * v_n).
20. x_n = \cos(u_n - v_n),
                                             y_n = \exp(u_n) * \cos(v_n).
21. x_n = \sin(\sin(u_n)),
                                             y_n = \sin(u_n) * \exp(v_n).
22. x_n = \sin(u_n + v_n),
                                             y_n = \cos(\sin(u_n + v_n)).
23. x_n = \sin(\cos(\sin(u_n))),
                                             y_n = u_n * v_n + \cos(v_n).
24. x_n = (\sin(u_n) + \cos(v_n)) * \exp(u_n), y_n = \exp(\cos(v_n)).
25. x_n = \exp(\cos(u_n + v_n)),
                                             y_n = \cos(u_n * v_n).
```

#### Примерная структура билетов тестирования

#### Тест «Программирование параллельных вычислительных процессов»

**Вопрос 1**. Установить соответствие между законами повышения производительности вычислительной системы и их описаниями.

Законы: 1) закон Мура; 2) закон Гроша; 3) гипотеза Минского; 4) закон Амдала.

*Описания*: 1) если доля последовательно выполняющихся в программе равна f, то ускорение параллельного вычисления этой программы не превышает 1/f;

- 2) в параллельной системе с п процессорами, производительность каждого из которых равна единице, общая производительность растет как  $log_{2}n$ ;
- 3) производительность одного процессора увеличивается пропорционально квадрату его стоимости;
- 4) количество транзисторов на кристалле процессора и производительность процессора удваивается каждые полтора года.

**Вопрос 2**. Установить соответствие между функциями классами классификации Флинна и примерами вычислительных систем, принадлежащим к этим классам.

*Функции*: 1) SISD; 2) SIMD); 3) MISD; 4) MIMD с локальной памятью; MIMD с общей памятью.

Назначение: 1) симметричная мультипроцессорная система SMP;

- 2) система с массовым параллелизмом МРР;
- 3) конвейерная система;
- 4) матричный процессор;
- 5) обычный компьютер.

**Bonpoc 3**. Установить соответствие между функциями Windows API и их назначением.

Функции: 1) CreateProcess();

- 2) CreateThread();
- 3) TerminateProcess();
- 4) TerminateThread().

Назначение: 1) принудительное завершение потока;

- 2) принудительное завершение процесса;
- 3) загрузка потока;
- 4) загрузка процесса.

Вопрос 4. Какое расширение имеет загружаемый процесс:

#### Варианты ответа:

1. exe; 2. dll; 2. cpp; 4. com.

#### Тест «Методы распараллеливания рекурсивных подпрограмм»

**Bonpoc 1**. Какой из приведенных ниже прототипов функции загружаемого потока описан верно?

#### Варианты ответа:

- 1) DWORD WINAPI thr(LPVOID p);
- 2) int thr(void \*p);
- 3) int thr(int a, int b);
- 4) DWORD WINAPI thr(int a).

Вопрос 2. Каким образом передаются аргументы подпрограммы загружаемого потока?

#### Варианты ответа:

- 1) определяют структуру, состоящую из аргументов подпрограммы и передают эти аргументы с помощью указателя на структуру;
- 2) как аргументы обычной подпрограммы;
- 3) определяют структуру, состоящую из аргументов подпрограммы и передают эту структуру как аргумент подпрограммы потока;
- 4) используют дополнительные средства операционой системы.

Вопрос 3. Каким образом распараллеливается вычисление ассоциативной операции над элементами массива.

#### Варианты ответа:

1) определяется рекурсивная подпрограмма, вычисляющая результат операции над элементами массива, затем рекурсивные вызовы в подпрограмме заменяются на загрузку потока подпрограммы, в главной программе подпрограмма вызывается без загрузки потока;

- 2) определяется рекурсивная подпрограмма, вычисляющая результат операции над элементами массива, затем рекурсивные вызовы в подпрограмме и в главной программе заменяются на загрузку потока подпрограммы;
- 3) определяется рекурсивная подпрограмма, вычисляющая результат операции над элементами массива, затем в подпрограмме потока оставляется рекурсивный вызов а в главной программе осуществляется загрузка потока подпрограммы.

#### Тест «Решения проблемы сериализации с помощью семафоров»

**Bonpoc 1**. Установить соответствие между функциями Windows API и их назначением.

Функции: 1) CreateSemaphore(); 2) CreateMutex(); 3) CreateEvent().

*Назначение*: 1) создание объекта события; 2) создание бинарного семафора; 3) создание семафора.

**Вопрос 2**. Установить соответствие между функциями Windows API и их назначением при работе с семафорами.

Функции: 1) WaitForSingleObject (); 2) ReleaseSemaphore ().

Назначение: 1) освобождение семафора; 2) захват семафора.

**Вопрос 3**. Решение какой задачи должно быть использовано для построения многопоточного приложения вычисляющего сумму элементов массива.

#### Варианты ответа:

- 1) проблема взаимного исключения;
- 2) задача о читателях и писателях;
- 3) задача о философах.

### Тест «Построение и программирование волновых систем с помощью сетей Петри»

**Bonpoc 1**. В каком из перечисленных ниже случаев срабатывание перехода сети Петри возможно и приводит к маркировке M'?

#### Варианты ответа:

- 1)  $M \ge pre(t)$ , M' = M pre(t) + post(t);
- 2)  $M \le pre(t)$ , M' = M pre(t) + post(t);
- 3)  $M \ge post$ , M' = M post(t) + pre(t);
- 4) M > pre(t), M' = M pre(t) + post(t).

**Вопрос 2**. Решение какой задачи используется для построения класса канала?

#### Варианты ответа:

- 1) задача о производителе и потребителе:
- 2) задача о читателях и писателях;
- 3) задача о философах.

**Вопрос 3**. Конвейерная система состоит из n потоков, соответствующих ее операциям. Синхронизация работы потоков осуществляется с помощью событий. Пусть n больше 10. Какое число k наиболее точно приближает количество событий, необходимых для синхронизации?

#### Варианты ответа:

1) 2n; 2) n; 3) 3n; 4) 4n.

**Вопрос 4**. Какое количество каналов необходимо для вычисления значений функции  $z_n = \exp(x_n) + x_n y_n$  с помощью волновой системы?

#### Варианты ответа:

- 1) 5; 2) 1; 3) 2; 4) 3; 5) 4.
- 8 Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины (модуля)
- 8.1 Основная литература
- 1 Максимов, Н. В. Архитектура ЭВМ и вычислительных систем [Электронный ресурс]: учебник / Н.В. Максимов, Т.Л. Партыка, И.И. Попов. 5-е изд., перераб. и доп. М. : Форум: НИЦ ИНФРА-М, 2015. 512с. // ZNANIUM.COM : электронно-библиотечная система. Режим доступа: http://www.znanium.com/catalog.php?.
- 2 Колдаев, В. Д. Основы алгоритмизации и программирования [Электронный ресурс]: учебное пособие / В.Д. Колдаев; под ред. Л.Г. Гагариной. М.: ИД ФОРУМ: ИНФРА–М, 2015. 416 с. // ZNANIUM.COM: электроннобиблиотечная система. Режим доступа: http://www.znanium.com/catalog.php?.
- 3 Кузин, А. В. Программирование на языке Си [Электронный ресурс] /А.В.Кузин, Е.В.Чумакова М.: Форум, НИЦ ИНФРА—М, 2015. 144с.// ZNANIUM.COM : электронно—библиотечная система. Режим доступа: http://www.znanium.com/catalog.php?.
- 4 Программная инженерия: учебник для вузов / под ред. Б.Г.Трусова. М.: Академия, 2014. 282с. (Высшее образование. Бакалавриат). Библиогр.: с.273–280.

#### 8.2 Дополнительная литература

1 Батчер, П. Семь моделей конкуренции и параллелизма за семь недель. Раскрываем тайны потоков / П. Батчер; пер. с англ. А. Н. Киселев. — М.: ДМК Пресс, 2015.-360c.

- 2 Гома, X. UML. Проектирование систем реального времени, параллельных и распределенных приложений / X. Гома; Пер. с англ. А.А.Слинкина. М.: ДМК Пресс, 2016. 698с. (Объектно-ориентированные технологии в программировании).
- 3 Грегори, К. С++ AMP: построение массивно параллельных программ с помощью Microsoft Visual C++ / К. Грегори, Э. Миллер. М.: ДМК Пресс, 2013.
- 4 Параллельные алгоритмы. Разработка и реализация: учебное пособие для вузов / Ю. К. Демьянович, И. Г. Бурова, Т. О. Евдокимова и др. М.: Интернет—университет информационных технологий: Бином. Лаборатория знаний, 2012. 343с. (Основы информационных технологий).

## 9 Перечень ресурсов информационно-телекоммуникационной сети «Интернет» (далее – сеть «Интернет»), необходимых для освоения дисциплины (модуля)

- 1 Информационно-аналитический центр по параллельному программированию http://www.parallel.ru .
  - 2 Национальный открытый университет ИНТУИТ http://www.intuit.ru.

### 10 Методические указания для обучающихся по освоению дисциплины (модуля)

Обучение дисциплине предполагает проведение аудиторных занятиях и выполнение студентом самостоятельной работы. Аудиторные занятия проходят в форме лекций и лабораторных занятий.

Во время лекционных занятий при написании конспекта лекций студенту рекомендуется кратко, схематично, последовательно фиксировать основные положения, формулировки, выводы, отмечать важные мысли, выделять ключевые слова, термины, делать пометки на вопросах, терминах, блоках в тексте, которые вызывают затруднения, после чего постараться найти ответ в рекомендуемой литературе. Если ответ не найден, то обратиться к преподавателю.

Во время лабораторных занятий студенту рекомендуется работать с конспектом лекций, использовать интернет-ресурсы для построения алгоритмов выполнения заданий лабораторной работы или контрольной работы. Также студенту необходимы знания принципов построения архитектуры ПО и видов архитектуры ПО при разработке многопоточных приложений. В случае затруднений, обратиться с вопросом к преподавателю.

Выполнение лабораторных работ и контрольной работы способствуют лучшему освоению практических навыков по данному предмету, закрепле-

ния и углубления навыков параллельного программирования. Студент получает задания в начале изучаемого раздела, а сдает выполненное задание после прохождения всех лабораторных занятий по данному разделу.

В рамках выполнения самостоятельной работы студент готовится к лабораторным занятиям, изучает и повторяет отдельные теоретические разделы дисциплины, выполняет и оформляет лабораторные работы и контрольную работу, а также готовится к их защите.

Текущий контроль учебной деятельности студентов осуществляется на лабораторных занятиях при тестировании и защите студентом лабораторных работ и контрольной работы. Проведение контроля текущей успеваемости, с одной стороны, позволяет получить достоверную информацию, как о степени освоения студентом теоретических разделов дисциплины, так и приобретения им практических навыков, с другой стороны, стимулирует ритмичность учебной деятельности. На тестирование выносятся практические задания, соответствующие всем теоретическим разделам дисциплины.

Защита лабораторных работ и контрольной работы проводится как в форме собеседования на лабораторном занятии, что позволяет определить уровень знаний студента основных понятий, алгоритмов и методов, так и в форме тестирования разработанного студентом ПО при различных входных данных и параметров, что позволяет оценить его знания принципов построения архитектуры ПО и видов архитектуры ПО, умения разрабатывать многопоточные приложения и навыки параллельного программирования.

11 Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения и информационных справочных систем (при необходимости)

Освоение дисциплины «Параллельное программирование» основывается на активном использовании студентом Microsoft Visual C++, интегрированной среде разработки приложений на языке C++.

С целью повышения качества ведения образовательной деятельности в университете создана электронная информационно-образовательная среда. Она подразумевает организацию взаимодействия между обучающимися и преподавателями через систему личных кабинетов студентов, расположенных на официальном сайте университета в информационнотелекоммуникационной сети «Интернет» по адресу https://student.knastu.ru.

Созданная информационно-образовательная среда позволяет контролировать ход образовательного процесса посредством размещения студентами в личных кабинетах отчетов о выполненной контрольной работе, проверкой преподавателем контрольной работы, по результатам которой либо РГР засчитывается, либо отправляется на доработку, но при этом преподаватель обязательно указывает конкретные замечания.

В процессе подготовки отчетов студентом активно используется программный продукт Microsoft Word.

Рекомендуемая среда разработки — распространяемая свободно (бесплатная) полнофункциональная интегрированная среда разработки Visual Studio Community 2017. Visual Studio Community может использовать неограниченное число пользователей в организации в следующих случаях: в учебных аудиториях, для научных исследований или участия в проектах с открытым кодом.

## 12 Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)

Для реализации программы дисциплины «Параллельное программирование» используется материально-техническое обеспечение, перечисленное в таблице 8.

Таблица 8 – Материально-техническое обеспечение дисциплины

Ауди тория	Наименова- ние аудито- рии (лаборато- рии)	Используемое оборудование	Назначение обору- дования
303-3,	компьютер-	10 персональных ЭВМ, каждая из которых	Проведение лабо-
305-3,	ные классы	оснащена процессором Intel(R) Core (TM)	раторных занятий,
312-3	ФКТ	i3-2100 CPU @3.10 GHz и оперативной па-	выполнение и за-
		мятью 2ГБ. Операционная система -	щита лаборатор-
		Windows 7. В классе имеется сетевой ком-	ных работ и кон-
		мутатор Cisco catalyst 2960 с ПО IOS ver	трольной работы
		12.2(55)SE5.	

### Лист регистрации изменений к РПД

<b>№</b> п/п	Содержание изменения/основание	Кол-во стр. РПД	Подпись автора РПД
1	Изменение листа подписей в связи со сменой декана ФКТ /пр.№ 271-ЛС «к» от 29.12.2016	1	
2	Изменение КУГ/пр. № 326-О «а» от 04.09.2017	7	
3	Изменение титульного листа в связи с переименованием вуза/пр. №997-О от 03.11.2017	1	
4	Актуализация литературы/ 20.11.2017	2	
5	Актуализация используемого ПО/ 28.11.2017	1	