

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Галунин Сергей Александрович
Должность: проректор по учебной работе
Дата подписания: 26.11.2024 14:26:37
Уникальный программный ключ:
08ef34338325bdb0ac5a47baa5472ce36cc3fc3b

Приложение к ОПОП
«Организация и программирова-
ние интеллектуальных систем»



СПбГЭТУ «ЛЭТИ»
ПЕРВЫЙ ЭЛЕКТРОТЕХНИЧЕСКИЙ

МИНОБРНАУКИ РОССИИ

федеральное государственное автономное образовательное учреждение высшего образования
**«Санкт-Петербургский государственный электротехнический университет
«ЛЭТИ» им. В.И.Ульянова (Ленина)»
(СПбГЭТУ «ЛЭТИ»)**

РАБОЧАЯ ПРОГРАММА

дисциплины

«ПАРАЛЛЕЛЬНЫЕ АЛГОРИТМЫ И СИСТЕМЫ»

для подготовки бакалавров

по направлению

09.03.01 «Информатика и вычислительная техника»

по профилю

«Организация и программирование интеллектуальных систем»

Санкт-Петербург

2024

ЛИСТ СОГЛАСОВАНИЯ

Разработчики:

доцент, к.т.н., доцент Пазников А.А.

Рабочая программа рассмотрена и одобрена на заседании кафедры ВТ
19.01.2024, протокол № 1

Рабочая программа рассмотрена и одобрена учебно-методической комиссией
ФКТИ, 24.01.2024, протокол № 1

Согласовано в ИС ИОТ

Начальник ОМОЛА Загороднюк О.В.

1 СТРУКТУРА ДИСЦИПЛИНЫ

Обеспечивающий факультет	ФКТИ
Обеспечивающая кафедра	ВТ
Общая трудоемкость (ЗЕТ)	3
Курс	4
Семестр	8

Виды занятий

Лекции (академ. часов)	16
Практические занятия (академ. часов)	16
Иная контактная работа (академ. часов)	1
Все контактные часы (академ. часов)	33
Самостоятельная работа, включая часы на контроль (академ. часов)	75
Всего (академ. часов)	120

Вид промежуточной аттестации

Лифф зачет (курс) 1

2 АННОТАЦИЯ ДИСЦИПЛИНЫ

«ПАРАЛЛЕЛЬНЫЕ АЛГОРИТМЫ И СИСТЕМЫ»

Курс "Параллельные алгоритмы и системы" представляет собой введение в теорию и практику разработки эффективных параллельных программ и систем, основанных на многопроцессорных архитектурах. Студенты изучат принципы параллельного программирования и алгоритмические подходы, необходимые для достижения масштабируемости, производительности и надежности в многопоточных средах. Будут рассмотрены различные модели памяти и проблемы, связанные с согласованностью кешей и взаимодействием потоков. Студенты изучат различные абстракции и примитивы параллельного программирования, такие как потоки, блокировки, семафоры и атомарные операции. Особое внимание будет уделено разработке параллельных алгоритмов для распределенных вычислений и обработки данных. В ходе практических занятий и проектных работ студенты будут решать задачи, требующие параллельного программирования, и анализировать их производительность и корректность. Курс призван подготовить студентов к эффективному использованию мощности многопроцессорных систем в различных областях, включая вычислительные науки, обработку больших данных и распределенные системы.

SUBJECT SUMMARY

«PARALLEL ALGORITHMS AND SYSTEMS»

The course "Parallel Algorithms and Systems" provides an introduction to the theory and practice of developing efficient parallel programs and systems based on multiprocessor architectures. Students will study the principles of parallel programming and algorithmic approaches necessary to achieve scalability, performance, and reliability in multithreaded environments. Various memory models and issues related to cache coherence and thread interaction will be examined. Students will explore

different abstractions and primitives of parallel programming, such as threads, locks, semaphores, and atomic operations. Special emphasis will be placed on developing parallel algorithms for distributed computing and data processing. Through practical exercises and project work, students will solve problems that require parallel programming and analyze their performance and correctness. The course aims to prepare students for effectively harnessing the power of multiprocessor systems in various fields, including computational sciences, big data processing, and distributed systems.

3 ОБЩИЕ ПОЛОЖЕНИЯ

3.1 Цели и задачи дисциплины

1. Цели изучения дисциплины:

- знакомство с основными концепциями и принципами параллельного программирования и разработки многопроцессорных систем;
- формирование практических навыков по разработке эффективных параллельных алгоритмов и программ для многопроцессорных архитектур;
- выработать понимание различных моделей памяти и проблем, связанных с параллельным выполнением и взаимодействием потоков;
- изучить основные абстракции и примитивы параллельного программирования, такие как потоки, блокировки, семафоры и атомарные операции;
- сформировать навыки анализа производительности и корректности параллельных программ, а также способность принимать решения о выборе и применении соответствующих алгоритмов и техник параллельного программирования;
- научиться эффективному использовать мощности многопроцессорных систем в различных областях, таких как вычислительные науки, обработка больших данных и распределенные системы.

2. Задачи изучения дисциплины:

- 1). Изучение основных понятий и теоретических основ параллельного программирования и многопроцессорных систем.
- 2). Изучение различных моделей памяти и их влияния на проектирование параллельных программ.
- 3). Освоение различных абстракций и примитивов параллельного программирования и их применение в разработке эффективных параллельных алгоритмов.
- 4). Формирование практических навыков в реализации и анализе производи-

тельности параллельных программ.

- 5). Формирование понимания проблем согласованности кешей, взаимодействия и синхронизации потоков в многопроцессорных системах.
- 6). Изучение алгоритмических подходов для параллельной обработки данных и распределенных вычислений.
- 7). Выработка умений выбирать и применять подходящие алгоритмы и техники параллельного программирования для различных задач.

3. Знания:

- основных концепций и принципов параллельного программирования и разработки многопроцессорных систем;
- различных моделей памяти и их влияния на разработку параллельных программ;
- абстракции и примитивы параллельного программирования, такие как потоки, блокировки, семафоры и атомарные операции.
- проблем согласованности кешей, взаимодействия и синхронизации потоков в многопроцессорных системах;
- алгоритмических подходов и техник параллельного программирования для эффективной обработки данных и распределенных вычислений;
- для анализа производительности и корректности параллельных программ;
- выбора и применения подходящих алгоритмов и техники параллельного программирования для различных задач и сред;
- основных принципов эффективного использования мощности многопроцессорных систем в различных областях, включая вычислительные науки, обработку больших данных и распределенные системы;

4. Умения:

- разрабатывать эффективные параллельные алгоритмы для различных задач и сред, учитывая особенности многопроцессорных систем;
- выбирать и применять соответствующие абстракции и примитивы параллель-

ного программирования, такие как потоки, блокировки, семафоры и атомарные операции;

-анализировать производительность и корректность параллельных программ, идентифицировать и устранять узкие места и проблемы синхронизации;

-работать с различными моделями памяти и эффективно управлять согласованностью кешей и взаимодействием потоков;

-разрабатывать параллельные алгоритмы для обработки больших объемов данных и распределенных вычислений, оптимизируя время выполнения и ресурсное использование;

-применять алгоритмические подходы и техники параллельного программирования для решения сложных задач в области вычислительных наук, обработки больших данных и распределенных систем;

-оценивать и эффективно использовать мощность многопроцессорных систем для достижения высокой производительности и масштабируемости в различных областях применения;

-принимать решения о выборе и применении соответствующих алгоритмов и техник параллельного программирования в зависимости от поставленных задач и требований производительности;

5. Навыки:

-разработки эффективных параллельных алгоритмов для различных задач и сред, учитывая особенности многопроцессорных систем;

-выбора и применения соответствующих абстракций и примитивов параллельного программирования, таких как потоки, блокировки, семафоры и атомарные операции, в том числе для задач искусственного интеллекта;

-анализа производительности и корректности параллельных программ, включая идентификацию и устранение узких мест и проблем синхронизации;

-работы с различными моделями памяти и управления согласованностью кешей и взаимодействием потоков;

- разработки параллельных алгоритмов для обработки больших объемов данных и распределенных вычислений, включая оптимизацию времени выполнения и использование ресурсов;
- применения алгоритмических подходов и техник параллельного программирования для решения сложных задач в области вычислительных наук, обработки больших данных и распределенных систем;
- оценки и эффективного использования мощности многопроцессорных систем для достижения высокой производительности и масштабируемости в различных областях применения;
- принятия решений о выборе и применении соответствующих алгоритмов и техник параллельного программирования в зависимости от поставленных задач и требований производительности.

3.2 Место дисциплины в структуре ОПОП

Дисциплина изучается на основе ранее освоенных дисциплин учебного плана:

1. «Программирование»
2. «Алгоритмы и структуры данных»
3. «Операционные системы»
4. «Сети ЭВМ»
5. «Архитектура вычислительных и информационных систем»
6. «Организация процессов и программирование в среде Linux»
7. «Основы компьютерного зрения»
8. «Производственная практика (научно-исследовательская работа)»

и обеспечивает подготовку выпускной квалификационной работы.

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

В результате освоения образовательной программы обучающийся должен достичь следующие результаты обучения по дисциплине:

Код компетенции/ индикатора компетенции	Наименование компетенции/индикатора компетенции
СПК-5	Способен разрабатывать программные компоненты компьютерных вычислительных систем и сетей, автоматизированных систем обработки информации
<i>СПК-5.1</i>	<i>Разрабатывает программные модули и компоненты</i>

4 СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

4.1 Содержание разделов дисциплины

4.1.1 Наименование тем и часы на все виды нагрузки

№ п/п	Наименование темы дисциплины	Лек, ач	Пр, ач	ИКР, ач	СР, ач
1	Введение в параллельные вычисления. Основные проблемы и задачи	1			
2	Взаимное исключение. Алгоритмы взаимного исключения	2	2		3
3	Разделяемые объекты. Линеаризуемость	2	2		4
4	Основы теории разделяемой памяти	2	3		4
5	Регистры. Консенсус. Протоколы консенсуса	2	3	1	20
6	Универсальность консенсуса	2	2		20
7	Алгоритмы реализации спинлоков	2	2		20
8	Алгоритмы реализации разделяемых связных списков	2	2		4
9	Алгоритмы реализации разделяемых очередей и стеков	1			
Итого, ач		16	16	1	75
Из них ач на контроль		0	0	0	0
Общая трудоемкость освоения, ач/зе		108/3			

4.1.2 Содержание

№ п/п	Наименование темы дисциплины	Содержание
1	Введение в параллельные вычисления. Основные проблемы и задачи	Краткий обзор курса. Цель курса. Задачи курса. Место курса в общем образовании бакалавра. Связь с другими дисциплинами
2	Взаимное исключение. Алгоритмы взаимного исключения	Определение FLOPS. Границы применимости FLOPS. Причины широкого распространения FLOPS. Параллельность вычислений. Развитие. Тенденции и проблемы. Трудности параллельного программирования.
3	Разделяемые объекты. Линеаризуемость	Пути достижения параллелизма. Устройство компьютера. Параллельная обработка данных. Конвейерная обработка данных. Технико-эксплуатационные характеристики ЭВМ. Характеристика типовых схем коммуникации в многопроцессорных вычислительных системах

№ п/п	Наименование темы дисциплины	Содержание
4	Основы теории разделяемой памяти	Классификации по признакам: принцип действия; используемая элементная база; назначение; размеры и вычислительная мощность; особенности архитектуры. Классификация Флинна. Векторные устройства. Классификация Ванга и Бриггса.
5	Регистры. Консенсус. Протоколы консенсуса	Система функциональных устройств. Модель вычислений в виде информационного графа алгоритма. Концепция неограниченного параллелизма. Определение времени выполнения параллельного алгоритма. Учебный пример: вычисление частных сумм последовательности числовых значений. Неформальная постановка задач параллельного программирования ВС. Граф-схемы параллельных алгоритмов.
6	Универсальность консенсуса	Алгоритмы оптимизации информационного графа по ширине и высоте. Определение групп вершин графа и минимальных высоты и ширины (приведение к параллельной форме). Оптимизация информационного графа с учетом межпроцессорных передач данных.
7	Алгоритмы реализации спинлоков	Оценка максимально достижимого параллелизма. Анализ масштабируемости параллельных вычислений. Временные оценки на информационных графах. Алгоритм нахождения ранних сроков окончания выполнения работ. Алгоритм нахождения поздних сроков окончания выполнения работ при заданном значении. Нижняя оценка минимального числа процессоров, необходимого для выполнения алгоритма за заданное время. Нижняя оценка минимального времени выполнения данного алгоритма на ВС.
8	Алгоритмы реализации разделяемых связных списков	Простые сети Петри. Маркировка сети. Моделирование простой вычислительной системы. Моделирование конструкции fork/join сетью Петри. Моделирование структуры parbegin/parend сетью Петри.
9	Алгоритмы реализации разделяемых очередей и стеков	Подведение итогов. Обзор специальных глав теории параллельных вычислений.

4.2 Перечень лабораторных работ

Лабораторные работы не предусмотрены.

4.3 Перечень практических занятий

Наименование практических занятий	Количество ауд. часов
1. Создание, компиляция и запуск параллельной программы	3
2. Организация обмена сообщениями между процессами	2
3. Параллельная реализация алгоритмов матричной алгебры.	2

Наименование практических занятий	Количество ауд. часов
4. Работа с коллективными функциями	2
5. Функции оценки параметров работы параллельных программ, функции блокировки и другие вспомогательные функции MPICH.	4
6. Распараллеливание численных методов	3
Итого	16

4.4 Курсовое проектирование

Курсовая работа (проект) не предусмотрены.

4.5 Реферат

Реферат не предусмотрен.

4.6 Индивидуальное домашнее задание

Индивидуальное домашнее задание не предусмотрено.

4.7 Доклад

Доклад не предусмотрен.

4.8 Кейс

Кейс не предусмотрен.

4.9 Организация и учебно-методическое обеспечение самостоятельной работы

Изучение дисциплины сопровождается самостоятельной работой студентов с рекомендованными преподавателем литературными источниками и информационными ресурсами сети Интернет.

Планирование времени для изучения дисциплины осуществляется на весь период обучения, предусматривая при этом регулярное повторение пройденного.

го материала. Обучающимся, в рамках внеаудиторной самостоятельной работы, необходимо регулярно дополнять сведениями из литературных источников материал, законспектированный на лекциях. При этом на основе изучения рекомендованной литературы целесообразно составить конспект основных положений, терминов и определений, необходимых для освоения разделов учебной дисциплины.

Особое место уделяется консультированию, как одной из форм обучения и контроля самостоятельной работы. Консультирование предполагает особым образом организованное взаимодействие между преподавателем и студентами, при этом предполагается, что консультант либо знает готовое решение, которое он может предписать консультируемому, либо он владеет способами деятельности, которые указывают путь решения проблемы.

Текущая СРС	Примерная трудоемкость, ач
Работа с лекционным материалом, с учебной литературой	20
Опережающая самостоятельная работа (изучение нового материала до его изложения на занятиях)	0
Самостоятельное изучение разделов дисциплины	20
Выполнение домашних заданий, домашних контрольных работ	0
Подготовка к лабораторным работам, к практическим и семинарским занятиям	20
Подготовка к контрольным работам, коллоквиумам	0
Выполнение расчетно-графических работ	0
Выполнение курсового проекта или курсовой работы	0
Поиск, изучение и презентация информации по заданной проблеме, анализ научных публикаций по заданной теме	0
Работа над междисциплинарным проектом	0
Анализ данных по заданной теме, выполнение расчетов, составление схем и моделей, на основе собранных данных	0
Подготовка к зачету, дифференциированному зачету, экзамену	15
ИТОГО СРС	75

5 Учебно-методическое обеспечение дисциплины

5.1 Перечень основной и дополнительной литературы, необходимой для освоения дисциплины

№ п/п	Название, библиографическое описание	К-во экз. в библ.
Основная литература		
1	Воеводин, Валентин Васильевич. Параллельные вычисления [Текст] : учеб. пособие для вузов по направлению 510200 "Прикладная математика и информатика" / В.В. Воеводин, Вл.В. Воеводин, 2002. -VII, 599 с.	12
2	Эндрюс, Грегори Р. Основы многопоточного, параллельного и распределенного программирования [Текст] : [Пер. с англ.] / Г.Р. Эндрюс, 2003. - 505 с.	20
3	Немнюгин, Сергей А. Параллельное программирование для многопроцессорных вычислительных систем [Текст] : монография / С.А.Немнюгин, О.Л.Стесик, 2002. -396 с.	10
Дополнительная литература		
1	Параллельные вычисления в локальных сетях [Текст] : метод. указания к лаб. работам по дисциплине "Параллельные алгоритмы и системы" / Санкт-Петербургский государственный электротехнический университет им. В.И. Ульянова (Ленина) "ЛЭТИ", 2005. -44 с.	83
2	Орлов, Сергей Александрович. Организация ЭВМ и систем [Текст] : учеб. для вузов по направлению подгот. дипломир. специалистов "Информатика и вычисл. техника" / С. А. Орлов, Б. Я. Цилькер, 2011. -686 с.	15

5.2 Перечень ресурсов информационно-телекоммуникационной сети «Интернет», используемых при освоении дисциплины

№ п/п	Электронный адрес
1	Cit Forumhttp://www.citforum.ru/
2	ИНТУИТ Национальный открытый университет. Параллельные вычисления и многопоточное программированиehttp://www.intuit.ru/studies/courses/10554/1092/info

5.3 Адрес сайта курса

Адрес сайта курса: <https://vec.etu.ru/moodle/course/view.php?id=14841>

6 Критерии оценивания и оценочные материалы

6.1 Критерии оценивания

Для дисциплины «Параллельные алгоритмы и системы» предусмотрены следующие формы промежуточной аттестации: зачет с оценкой.

Зачет с оценкой

Оценка	Описание
Неудовлетворительно	Курс не освоен. Студент испытывает серьезные трудности при ответе на ключевые вопросы дисциплины
Удовлетворительно	Студент в целом овладел курсом, но некоторые разделы освоены на уровне определений и формулировок
Хорошо	Студент овладел курсом, но в отдельных вопросах испытывает затруднения. Умеет решать задачи
Отлично	Студент демонстрирует полное овладение курсом, способен применять полученные знания при решении конкретных задач

Особенности допуска

Для допуска к дифф. зачету нужно сдать 3 практические работы на положительную оценку ("удовлетворительно" и выше).

Правила проведения дифференцированного зачета по дисциплине "Параллельные алгоритмы и системы" предназначены для оценки уровня усвоения студентами теоретических и практических аспектов параллельного программирования и разработки систем на многопроцессорных архитектурах. Зачет с оценкой включает в себя следующие критерии:

1. Теоретическое Понимание: Студенты должны продемонстрировать глубокое понимание основных принципов параллельного программирования, моделей памяти и проблем согласованности кешей.
2. Практические Навыки: Оценивается способность студентов разрабатывать параллельные алгоритмы и программы, использовать абстракции параллельного программирования, а также проводить анализ производительности и корректности разработанных решений.
3. Проектные Работы: Студенты должны успешно решить задачи, требующие параллельного программирования, в рамках практических занятий и проектов. Оценивается качество реализации и адаптация решений к конкретным требованиям.
4. Теоретический Анализ: Студенты должны способствовать обоснованному анализу выбора алгоритмов, подходов к параллельному программированию и решениям проблем согласованности.
5. Знание Профессиональной Терминологии: Оценивается уровень знания ключевых терминов и понятий, связанных с параллельными алгоритмами и системами.

Оценка будет выставляться по 4-балльной шкале: от "неудовлетворительно" до "отлично". Каждый из указанных критериев будет вносить свой вклад в итоговую оценку.

6.2 Оценочные материалы для проведения текущего контроля и промежуточной аттестации обучающихся по дисциплине

Вопросы к дифф.зачету

№ п/п	Описание
1	Критерии корректности параллельных программ
2	Раскрыть понятие консенсуса. В чем состоит значение консенсуса?
3	Описать универсальные конструкции на базе консенсуса
4	Описать алгоритмы спинлоков
5	Какие существуют методы построения потокобезопасных структур, на примере списка?
6	Описать алгоритмы построения потокобезопасных очередей и стеков
7	В чем преимущества неблокируемой синхронизации?
8	Какие существуют методы повышения пропускной способности разделяемых структур данных?
9	Описать алгоритмы построения разделяемого связного списка на основе блокировок и без блокировок
10	Описать технические аспекты построения кэш-памяти
11	Какие существуют виды регистров?
12	Описать алгоритмы атомарных снимков
13	Алгоритмы разделяемых потокобезопасных очередей
14	Алгоритмы разделяемых потокобезопасных стеков
15	Методы повышения эффективности потокобезопасных стеков
16	Блокируемые реализации разделяемого потокобезопасного связного списка, реализующего множество
17	Неблокируемые реализации разделяемого потокобезопасного связного списка, реализующего множество
18	Потокобезопасные хеш-таблицы с открытой адресацией
19	Потокобезопасные хеш-таблицы с закрытой адресацией
20	Алгоритмы реализации масштабируемых потокобезопасных пулов
21	Алгоритмы реализации масштабируемых потокобезопасных сетей подсчета (counting network)
22	Алгоритмы барьерной синхронизации
23	Алгоритмы реализации программной транзакционной памяти
24	Критерии согласованности в многопоточных программах

Весь комплект контрольно-измерительных материалов для проверки сформированности компетенции (индикатора компетенции) размещен в закрытой части по адресу, указанному в п. 5.3

6.3 График текущего контроля успеваемости

Неделя	Темы занятий	Вид контроля
1	7. Алгоритмы реализации спинлоков	
2		
3		
4		Практическая работа
5	8. Алгоритмы реализации разделяемых связных списков	
6		Практическая работа
7	9. Алгоритмы реализации разделяемых очередей и стеков	
8		Практическая работа

6.4 Методика текущего контроля

на лекционных занятиях

Текущий контроль включает в себя контроль посещаемости (не менее **80** % занятий), по результатам которого студент получает допуск на ДЗ.

на практических (семинарских) занятиях

Текущий контроль включает в себя контроль посещаемости (не менее **80** % занятий), по результатам которого студент получает допуск на ДЗ.

В ходе проведения семинарских и практических занятий целесообразно привлечение студентов к как можно более активному участию в дискуссиях, решении задач, обсуждениях и т. д. При этом активность студентов также может учитываться преподавателем, как один из способов текущего контроля на практических занятиях.

выполнение практических работ

Текущий контроль включает в себя проверку корректности выполнения практических работ. Студенту выдается 3 практических работы, за выполнение которых он получает оценку. Для допуска к ДЗ, студент должен получить как минимум оценку "удовлетворительно" по каждой из практических работ.

Критерии оценивания работ:

- "Отлично" (5 баллов): Глубокое понимание, творческий подход, качествен-

ное исследование, профессиональное исполнение, расширение границ.

- ”Хорошо” (4 балла): Хорошее понимание, элементы творчества, аргументированные выводы, аккуратное исполнение.
- ”Удовлетворительно” (3 балла): Базовое понимание, минимальное творчество, ограниченный анализ, минимальное исполнение.
- ”Неудовлетворительно” (2 балла): Отсутствует базовое понимание, не проведен анализ, отсутствует фактическое исполнение.

самостоятельной работы студентов

Контроль самостоятельной работы студентов осуществляется на лекционных и практических занятиях студентов по методикам, описанным выше.

7 Описание информационных технологий и материально-технической базы

Тип занятий	Тип помещения	Требования к помещению	Требования к программному обеспечению
Лекция	Лекционная аудитория	Количество посадочных мест – в соответствии с контингентом, рабочее место преподавателя, маркерная доска, видеопроектор или мультимедиа-система, ПК	1) Операционная система GNU/Linux 3) Компиляторы clang, gcc 3) Стандартный стек ПО GNU/Linux
Практические занятия	Аудитория	Количество посадочных мест – в соответствии с контингентом, рабочее место преподавателя, проектор, экран, маркерная доска, ПК в количестве $\frac{1}{2}$ контингента	1) Операционная система GNU/Linux 3) Компиляторы clang, gcc 3) Стандартный стек ПО GNU/Linux
Самостоятельная работа	Помещение для самостоятельной работы	Оснащено компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду университета.	1) Операционная система GNU/Linux 3) Компиляторы clang, gcc 3) Стандартный стек ПО GNU/Linux

8 Адаптация рабочей программы для лиц с ОВЗ

Адаптированная программа разрабатывается при наличии заявления со стороны обучающегося (родителей, законных представителей) и медицинских показаний (рекомендациями психолого-медико-педагогической комиссии). Для инвалидов адаптированная образовательная программа разрабатывается в соответствии с индивидуальной программой реабилитации.

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

№ п/п	Дата	Изменение	Дата и номер протокола заседания УМК	Автор	Начальник ОМОЛА