

Документ подписан простой электронной подписью  
Информация о владельце:  
ФИО: Галунин Сергей Александрович  
Должность: проректор по учебной работе  
Дата подписания: 23.12.2025 12:07:09  
Уникальный программный ключ:  
08ef34338325bdb0ac5a47baa5472ce36cc3fc3b

Приложение к ОПОП  
«Информационно-управляющие  
системы»



**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

**«ОСНОВЫ ФРОНТЕНД-РАЗРАБОТКИ Ч.1»**

для подготовки бакалавров

по направлению

09.03.02 «Информационные системы и технологии»

по профилю

**«Информационно-управляющие системы»**

Санкт-Петербург

2025

## **ЛИСТ СОГЛАСОВАНИЯ**

Разработчики:

доцент, к.т.н., доцент Беляев С.А.

Рабочая программа рассмотрена и одобрена на заседании кафедры МОЭВМ  
20.01.2025, протокол № 1

Рабочая программа рассмотрена и одобрена учебно-методической комиссией  
ФКТИ, 28.01.2025, протокол № 1

Согласовано в ИС ИОТ

Начальник ОМОЛА Загороднюк О.В.

## **1 СТРУКТУРА ДИСЦИПЛИНЫ**

Обеспечивающий факультет	ФКТИ
Обеспечивающая кафедра	МОЭВМ
Общая трудоемкость (ЗЕТ)	3
Курс	3
Семестр	5

### **Виды занятий**

Электронные лекции (акад. часов)	34
Электронные практические (академ. часов) (академ. часов)	34
Иная контактная работа (академ. часов)	1
Все контактные часы (академ. часов)	1
Самостоятельная работа, включая часы на контроль (академ. часов)	39
Всего (академ. часов)	108

### **Вид промежуточной аттестации**

Дифф. зачет (курс)	3
--------------------	---

## **2 АННОТАЦИЯ ДИСЦИПЛИНЫ**

### **«ОСНОВЫ ФРОНТЕНД-РАЗРАБОТКИ Ч.1»**

Дисциплина является первой частью курса, который посвящен основам, необходимым для разработчиков интернет-сайтов и web-приложений. Рассматриваются основы построения web-приложений с использованием HTML, CSS. Обучающиеся последовательно изучают основы протокола HTTP, настройку web-сервера, основы JavaScript, TypeScript, сборку кода на Vite. Обсуждаются вопросы разработки клиентских приложений с использованием библиотеки React. Обучающиеся получают навыки тестирования React-приложений.

#### **SUBJECT SUMMARY**

#### **«FUNDAMENTALS OF FRONTEND DEVELOPMENT PART 1»**

The discipline is the first part of the course, which focuses on the basics necessary for developers of websites and web applications. The basics of building web applications using HTML and CSS are considered. Students consistently learn the basics of the HTTP protocol, configuring a web server, the basics of JavaScript, TypeScript, and building code on Vite. The issues of developing client applications using the React library are discussed. Students gain skills in testing React applications.

## **3 ОБЩИЕ ПОЛОЖЕНИЯ**

### **3.1 Цели и задачи дисциплины**

1. Целью дисциплины является приобретение теоретических знаний о современных web-технологиях и фреймворках и практических умений и навыков по всем этапам разработки web-приложений, включая разработку web-серверов, создания и оформления web-клиентов, тестирования приложений.
2. Задачами изучения дисциплины является приобретение теоретических знаний по проектированию web-приложений, формирование практических умений и навыков по разработке серверной и клиентской части web-приложений, оформлению и тестированию приложений.
3. Обучающиеся приобретают знания основных подходов по построению клиентской и серверной части web-приложений, по оформлению web-страниц, в том числе для просмотра с мобильных устройств, по решению типовых задач тестирования, сборки и обеспечения безопасности web-приложений с использованием языков JavaScript, TypeScript.
4. В результате изучения курса приобретаются умения разрабатывать клиентскую и серверную часть web-приложений, осуществлять настройку, тестирование и автоматизированную сборку web-приложений.
5. У обучающихся формируются навыки создания статических HTML-страниц, динамических web-приложений на языках JavaScript и TypeScript, навыки применения современных фреймворков разработки клиентских и серверных частей web-приложений.

### **3.2 Место дисциплины в структуре ОПОП**

Дисциплина изучается на основе ранее освоенных дисциплин учебного плана:

1. «Информатика»
2. «Программирование»
3. «Информационные технологии»
4. «Алгоритмы и структуры данных»

и обеспечивает изучение последующих дисциплин:

1. «Основы фронтенд-разработки ч.2»
2. «Архитектура вычислительных сетей»
3. «Производственная практика (преддипломная практика)»

### **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:

<b>Код компетенции/ индикатора компетенции</b>	<b>Наименование компетенции/индикатора компетенции</b>
ПК-0	Способен разрабатывать информационные модели и применять их для решения задач профессиональной деятельности
ПК-0.1	<i>Знает современные виды информационных моделей, применяемых при решении задач профессиональной деятельности</i>
ПК-0.2	<i>Создает и модифицирует информационные модели для решения задач профессиональной деятельности</i>
ПК-0.3	<i>Применяет информационные модели для решения задач профессиональной деятельности</i>

## **4 СОДЕРЖАНИЕ ДИСЦИПЛИНЫ**

### **4.1 Содержание разделов дисциплины**

#### **4.1.1 Наименование тем и часы на все виды нагрузки**

<b>№ п/п</b>	<b>Наименование темы дисциплины</b>	<b>ЭЛек, ач</b>	<b>ЭПр, ач</b>	<b>ИКР, ач</b>	<b>СР, ач</b>
1	Введение	1			
2	Что такое HTML и CSS	8	8		8
3	Дизайн, адаптивная вёрстка, декорирование	8	8		8
4	JavaScript	8	9		8
5	TypeScript	8	9		8
6	Заключение	1		1	7
	Итого, ач	34	34	1	39
	Из них ач на контроль	0	0	0	0
	Общая трудоемкость освоения, ач/зе				108/3

#### **4.1.2 Содержание**

<b>№ п/п</b>	<b>Наименование темы дисциплины</b>	<b>Содержание</b>
1	Введение	Рабочая программа, коммуникации между заинтересованными лицами.
2	Что такое HTML и CSS	Задачи frontend-разработчиков, базовые элементы, HTML, CSS. Теги HTML, заголовки, абзац, ссылка, атрибуты, один тег в другом, изображения, структура HTML-документа, правила CSS, тег style, CSS-файл, блоки, отступы, подпись к обложке, playground. Установка и настройка Git, SSH-ключ, настройка аккаунта GitHub, основы командной строки Bash, основные команды для организации командной работы с Git в локальных и удалённых репозиториях. Подключение шрифтов к странице, настройки шрифтов, подключение внешних шрифтов, оформление текста, переполнение текстовых блоков. Семантика HTML5, структура документа, оптимизация вёрстки, использование идентификаторов. Статическое, относительное и абсолютное позиционирование, z-index, фиксированное и «липкое» позиционирование, outline, стилизация полей ввода, кнопок и ярлыков, псевдо-классы валидации, выпадающие списки, кастомные элементы ввода.

<b>№ п/п</b>	<b>Наименование темы дисциплины</b>	<b>Содержание</b>
3	Дизайн, адаптивная вёрстка, декорирование	Дизайн в мире множества устройств, разница между резиновой и адаптивной вёрсткой, адаптивный макет, синтаксис медиа-запросов, характеристики устройств, пользовательские предпочтения, синтаксис диапазонов, выражения от контейнера. Форматы, оптимизация изображений, плотность пикселей, адаптивный выбор изображений из набора, image-set(), loading="lazy", относительные размеры блоков и текстовых элементов, функции calc(), min(), max(), clamp(). Экспорт SVG, использование SVG: онлайн, <img>, фон, use, fill, stroke, анимация и оптимизация SVG, SVG-маски. 2D-и 3D-трансформации, анимации, декорирование.
4	JavaScript	Примитивные типы данных, операторы typeof, undefined и null, строки, числа и специальные числовые значения. DOM: выбор элементов, атрибуты и их методы, манипуляции с классами CSS, свойства textContent, события, методы insertAdjacentHTML и insertAdjacentText. Чтение ошибок, их типы, логические ошибки и console.log, поиск документации, отладка через debugger, conditional breakpoints. Let и const против var, методы поиска в строке, преобразования строк и работы с числами, явное и неявное преобразование типов, логические операторы, НЕ (!), ИЛИ (  ), И (&&), XOR (^), switch-case, тернарный оператор, циклы, директивы break и continue. InnerHTML, createElement и createTextNode, добавление элементов на страницу, удаление и перемещение, клонирование, template-элементы, объект event, родственные связи в DOM. Объединение и преобразование в строку, управление элементами на любых позициях, коллекции в DOM, методы forEach и map, функции обратного вызова и их аргументы, методы filter, some, every, find, reduce, сортировка массива. Область видимости функции, rest и spread, деструктурирование аргументов. Создание объектов и запись свойств, прототипное наследование, дескрипторы свойств, деструктурирование массивов и объектов. Модули в JS, асинхронность, обработка событий.

<b>№ п/п</b>	<b>Наименование темы дисциплины</b>	<b>Содержание</b>
5	TypeScript	Динамическая и статическая типизация, документация JSDoc, проверка типов во время исполнения, настройка и инструментарий TS, массивы и объекты, специальные типы, создание собственных типов. Источники контекста, this, способы привязки, синтаксис классов в JS, добавление и типизация полей и методов классов, интерфейсы классов, принципы в ООП, инкапсуляция, наследование, полиморфизм. Приведение типов, типизация DOM-элементов и их событий, типизация стандартных объектов JS, файлы деклараций .d.ts, создание и использование библиотек. Статические поля и методы, имплементация интерфейсов, дженерики, условные типы. Композиция классов, паттерны singleton, adapter/facade/proxy, observer, builder. IntersectionObserver, ResizeObserver, FileReader, локальные хранилища. Дропдаун, табы, аккордеон, слайдер, галерея, уведомления, календарь, игра морской бой.
6	Заключение	Общие выводы по изложенному материалу. Содержание второй части курса. Связь с последующими дисциплинами траектории.

#### **4.2 Перечень лабораторных работ**

Лабораторные работы не предусмотрены.

#### **4.3 Перечень практических занятий**

<b>Наименование практических занятий</b>	<b>Количество ауд. часов</b>
1. Вёрстка	8
2. Анимация, шрифты, фреймы	8
3. Валидация форм и инфраструктура	8
4. Взаимодействие с сервером и GitHub Pages	10
Итого	34

#### **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**

Самостоятельной записи на курс нет. Студент заходит на курс, используя логин/пароль от единой учетной записи университета (единий логин и пароль). Каждую неделю будет доступна новая тема курса: кратко раскрывающие содержание каждой темы, презентации и конспекты, с которыми обучающиеся смогут ознакомиться в любое удобное время. Все темы включают практические занятия, которые предусматривают самостоятельное выполнение заданий, а также задания с автоматической проверкой, результаты которых учитываются при общей аттестации полученных знаний. В конце каждой лекции необходимо пройти небольшой контрольный тест, который покажет насколько усвоен предложенный материал. Рекомендуем изучать материал последовательно, что существенно облегчит работу. У каждого контрольного задания имеется своя форма (тест или практическое задание) есть срок выполнения (окончательный срок), по истечении которого даже правильные ответы система принимать не будет! В расписании курса указан окончательный срок каждого задания, который варьируется от двух до четырех недель в зависимости от его сложности. Весь учебный курс рассчитан на 16 недель. Его итоги будут подведены в тече-

ние нескольких недель после его окончания.

Текущая СРС	Примерная трудоемкость, ач
Работа с лекционным материалом, с учебной литературой	13
Опережающая самостоятельная работа (изучение нового материала до его изложения на занятиях)	8
Самостоятельное изучение разделов дисциплины	0
Выполнение домашних заданий, домашних контрольных работ	0
Подготовка к лабораторным работам, к практическим и семинарским занятиям	10
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, дифференцированному зачету, экзамену	8
<b>ИТОГО СРС</b>	<b>39</b>

## **5 Учебно-методическое обеспечение дисциплины**

### **5.1 Перечень основной и дополнительной литературы, необходимой для освоения дисциплины**

<b>№ п/п</b>	<b>Название, библиографическое описание</b>	<b>К-во экз. в библ.</b>
<b>Основная литература</b>		
1	Беляев С. А. Разработка игр на языке JavaScript : учебное пособие для вузов / С. А. Беляев, 2023. -152 с. -Текст : электронный.	неогр.
2	Беляев, Сергей Алексеевич. Web-технологии : электрон. лаб. практикум / С. А. Беляев, 2019. -1 эл. опт. диск (CD-ROM). -Текст : электронный.	неогр.
<b>Дополнительная литература</b>		
1	Дронов В. А. Nodejs, Express, MongoDB и React : 23 урока для начинающих / В. А. Дронов, 2024. -608 с. -Текст : непосредственный.	неогр.

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

<b>№ п/п</b>	<b>Электронный адрес</b>
1	React. JavaScript-библиотека для создания пользовательских интерфейсов <a href="https://ru.reactjs.org/">https://ru.reactjs.org/</a>

### **5.3 Адрес сайта курса**

Адрес сайта курса: <https://vec.etu.ru/moodle/course/view.php?id=23641>

## **6 Критерии оценивания и оценочные материалы**

### **6.1 Критерии оценивания**

Для дисциплины «Основы фронтенд-разработки ч.1» формой промежуточной аттестации является дифф. зачет. Оценивание качества освоения дисциплины производится с использованием рейтинговой системы.

#### **Дифференцированный зачет**

<b>Оценка</b>	<b>Количество баллов</b>	<b>Описание</b>
Неудовлетворительно	0 – 7	теоретическое содержание курса не освоено, необходимые практически навыки и умения не сформированы, выполненные учебные задания содержат грубые ошибки, дополнительная самостоятельная работа над курсом не приведет к существенному повышению качества выполнения учебных заданий
Удовлетворительно	8 – 11	теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые практические навыки и умения работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки
Хорошо	12 – 13	теоретическое содержание курса освоено полностью, без пробелов, некоторые практические навыки и умения сформированы недостаточно, все предусмотренные программой обучения учебные задания выполнены, качество выполнения ни одного из них не оценено минимальным числом баллов, некоторые виды заданий выполнены с ошибками
Отлично	14 – 15	теоретическое содержание курса освоено полностью, без пробелов, необходимые практические навыки и умения сформированы, все предусмотренные программой обучения учебные задания выполнены, качество их выполнения оценено количеством баллов, близким к максимальному

## **Особенности допуска**

К дифференцированному зачету допускаются студенты, показавшие прогресс в ЭОС Moodle (не менее 80%) и выполнившие 3 практические работы на оценку не менее "удовлетворительно".

Дифференцированный зачет проводится в виде тестирования в электронной образовательной среде Moodle. Тест содержит 15 вопросов с развернутыми ответами. Каждый вопрос теста оценивается в 1 балл в случае правильного ответа. Оценка за дифф. зачет выставляется в зависимости от количества баллов, полученных за итоговый тест следующим образом:

"Неудовлетворительно" -менее 8 баллов.

"Удовлетворительно" -от 8 до 11 баллов.

"Хорошо" -от 12 до 13 баллов.

"Отлично" -от 14 до 15 баллов.

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

### **Вопросы к дифф.зачету**

<b>№ п/п</b>	<b>Описание</b>
1	Структура HTML□документа
2	Настройка аккаунта GitHub
3	Основные команды для организации командной работы с Git в локальных и удалённых репозиториях
4	Семантика HTML5, структура документа, оптимизация вёрстки, использование идентификаторов
5	Правила CSS, тег style, CSS□файл
6	Разница между резиновой и адаптивной вёрсткой
7	Форматы, оптимизация изображений, плотность пикселей
8	Функции calc(), min(), max(), clamp()
9	Экспорт SVG
10	2D-и 3D-трансформации, анимации, декорирование
11	DOM: выбор элементов, атрибуты и их методы, манипуляции с классами CSS
12	Методы insertAdjacentHTML и insertAdjacentText
13	innerHTML, createElement и createTextNode
14	Методы forEach и map

15	Область видимости функции, rest и spread, деструктурирование аргументов
16	Модули в JS, асинхронность, обработка событий
17	Динамическая и статическая типизация, документация JSDoc
18	Инкапсуляция, наследование, полиморфизм
19	Статические поля и методы
20	IntersectionObserver, ResizeObserver, FileReader

## Вариант теста

1. Что такое замыкание (closure) в JavaScript?
  - a) Функция, которая выполняется немедленно.
  - b) Функция, которая имеет доступ к переменным внешней (окружающей) функции, даже после того, как внешняя функция завершила свое выполнение.**
  - c) Новый тип данных в JavaScript.
  - d) Объект, который хранит только статические данные.
2. В чем разница между == и === в JavaScript?
  - a) == выполняет строгое сравнение, а === – нет.
  - b) === выполняет строгое сравнение, а == – нет.**
  - c) == используется для сравнения чисел, а === – для строк.
  - d) Разницы нет.
3. Что выведет в консоль следующий код: console.log(3 + 2 + "7"); ?
  - a) "57"**
  - b) 12
  - c) "327"
  - d) 127
4. Что такое прототипное наследование в JavaScript?
  - a) Способ создания новых классов.

**b) Механизм, позволяющий объектам наследовать свойства и методы от других объектов.**

- c) Способ создания копий объектов.
- d) Новый синтаксис для создания функций.

5. Что такое hoisting в JavaScript?

a) Процесс оптимизации кода.

**b) Механизм, при котором объявления переменных и функций перемещаются в начало области видимости до начала выполнения кода.**

- c) Способ обработки ошибок.
- d) Новый синтаксис для создания циклов.

6. Что такое spread operator (...) в JavaScript?

a) Оператор для создания копий объектов.

**b) Оператор, позволяющий расширять итерируемые объекты (например, массивы) в местах, где ожидается несколько аргументов или элементов.**

- c) Оператор для создания прототипов.
- d) Оператор для выполнения асинхронных операций.

7. Что выведет в консоль следующий код?

```
const arr = [1, 2, 3];
```

```
arr.map(x => x * 2);
```

```
console.log(arr);
```

- a) [2, 4, 6]

**b) [1, 2, 3]**

- c) undefined

d) Ошибка

8. Что такое JSON?

a) Язык программирования.

**b) Формат обмена данными, основанный на текстовом представлении объектов JavaScript.**

c) База данных.

d) Метод шифрования данных.

9. Что выведет в консоль следующий код?

```
console.log(1 == '1');
```

```
console.log(1 === '1');
```

a) true, true

b) false, false

**c) true, false**

d) false, true

10. Что такое arrow function (`=>`) в JavaScript?

a) Новый тип цикла.

**b) Более короткий синтаксис для написания функций.**

c) Функция обратного вызова.

d) Способ объявления переменных.

## Образцы задач (заданий) для контрольных (проверочных) работ

### Пример практической работы

Требуется создать одностраничный сайт. В этом проекте вы соберёте его первую часть, затронув флексбокс-вёрстку, позиционирование и продвинутую семантику языка HTML. Для наполнения сайта мы выбрали материал о том,

как нужно учиться. Верстая текст и картинки, вы познакомитесь с методиками, которые помогут сделать ваш образовательный путь эффективным и захватывающим.

Откройте бриф с описаниями и иллюстрациями. На его первой странице представлен образ итогового результата. На остальных страницах — разбор секций сайта, размеры и отступы каждого элемента.

Скачайте архив с проектом. Делайте проектную работу в локальном репозитории, отправляйте файлы к себе в удалённый репозиторий.

В архиве для вас:

- Создана нужна файловая структура.
- Собраны все нужные для проекта картинки.
- Написан код, который устанавливает веб-странице определённые размеры.
- Даны дополнительные комментарии к уже написанному коду — они помогут лучше понять некоторые нюансы проекта.
- В корне лежит файл .editorconfig с настройками для текстового редактора.

Когда сочтёте, что работа выполнена, проверьте себя по чек-листву, через валидаторы, поместите код в git.

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**

<b>Неделя</b>	<b>Темы занятий</b>	<b>Вид контроля</b>
1	Что такое HTML и CSS Дизайн, адаптивная вёрстка, декорирование	Практическая работа
2		
3		
4		
5		
6	JavaScript	Практическая работа
7		
8		
9		
10		
11	TypeScript	Практическая работа
12		
13		
14		
15		
16		

### **6.4 Методика текущего контроля**

#### **на лекционных занятиях**

Текущий контроль включает в себя:

- контроль прогресса в ЭОС Moodle (не менее 80%).

#### **на практических занятиях**

- контроль прогресса в ЭОС Moodle (не менее 80%)

В процессе обучения по дисциплине студент обязан выполнить 3 практические работы в ЭОС Moodle.

Студент должен показать: понимание постановки задачи, подхода к ее решению, ход решения, выбор тех или иных методик решения задачи. Преподаватель определяет глубину понимания теоретического материала, который лежит в основе решения задачи практической работы, а также самостоятельность ее выполнения.

**Критерии оценивания практических работ:**

«отлично» - вопрос раскрыт полностью, задача решена правильно

«хорошо» - вопрос раскрыт не полностью, задача решена частично

«удовлетворительно» - в ответе на вопрос имеются существенные ошибки; задача не решена или решена неправильно, ход решения правильный

«неудовлетворительно» - отсутствует ответ на вопрос или содержание ответа не совпадает с поставленным вопросом, задача не решена, ход решения неправильный

### **самостоятельной работы студентов**

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

## **7 Описание информационных технологий и материально-технической базы**

<b>Тип занятий</b>	<b>Тип помещения</b>	<b>Требования к помещению</b>	<b>Требования к программному обеспечению</b>
Лекция	Лекционная аудитория	Количество посадочных мест – в соответствии с контингентом, рабочее место преподавателя, ПК	1) Linux Альт Образование 10 и выше; 2) Р7-Офис 7 и выше либо LibreOffice 7 и выше
Практические занятия	Аудитория	Количество посадочных мест – в соответствии с контингентом, рабочее место преподавателя, ПК	1) Linux Альт Образование 10 и выше; 2) Р7-Офис 7 и выше либо LibreOffice 7 и выше
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) Linux Альт Образование 10 и выше; 2) Р7-Офис 7 и выше либо LibreOffice 7 и выше

## **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**

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