



Настоящая рабочая программа разработана в соответствии с требованиями Федерального государственного образовательного стандарта среднего общего образования и на основе требований к результатам освоения основной образовательной программы среднего общего образования физико-математической школы-интерната ФГАОУ ВО «Сибирский федеральный университет». Элективный курс «Разработка сетевых приложений» рассчитан на 68 часов в год, изучается в 11 классе по выбору обучающихся в объеме 2 часа в неделю.

Современные информационные системы по своей природе являются распределенными системами. Рабочие станции пользователей, серверы приложений, серверы баз данных и прочие сетевые узлы повсеместно используются во всех сферах организации различных информационных процессов. Поэтому современные ИТ-специалисты должны понимать принципы построения сетевых приложений различного уровня и осваивать методы их построения при решении практико-ориентированных задач.

Данная общеобразовательная общеобразовательная программа технической направленности позволяет обучающимся овладеть основами проектирования распределенных сетевых приложений, получить навыки администрирования систем на базе ОС Linux, освоить программирование на языке Go, познакомиться с основными технологиями, используемыми для построения распределенных систем (базы данных, кэширование, серверы очередей, балансировка нагрузки), научиться использовать современные средства разработки и отладки приложений. Освоение данной программы позволит учащимся получить представление о профессиях инженера-проектировщика и программиста-разработчика серверных приложений и поможет сделать выбор будущей профессии.

**Актуальность** программы обуславливается стремительным развитием современных сетевых технологий и средств для разработки серверных приложений, что требует обучения специалистов в данных отраслях.

**Новизна** программы заключается в знакомстве обучающихся с современными технологиями программирования, принципами разработки архитектуры серверных приложений, приемами и способами разработки и тестирования кода, которые позволят решать широкий спектр задач и эффективно продолжать дальнейшее самостоятельное обучение в данной области.

**Цель программы** – освоение обучающимися основных навыков проектирования и разработки современных сетевых приложений, знакомство с инструментарием и принципами профессионального программирования.

### **Задачи программы**

#### *Обучающие:*

- обучить основным принципам проектирования архитектуры современных микросервисных приложений;
- познакомить с техническими и программными средствами организации сетей передачи и обработки данных;
- формировать представления о современных инструментах и программных средствах для работы программиста;
- обучать эффективной работе с технической литературой, интернет источниками;
- формировать целостную научную картину мира.

#### *Развивающие:*

- развивать алгоритмическое мышление, умение структурировать данные и процессы;
- развивать мыслительные, творческие, коммуникативные способности обучающихся;

- развивать интерес к техническим знаниям;
- развивать интеллектуальные и практические умения, самостоятельно приобретать и применять на практике полученные знания;
- развивать умение осознанно ставить перед собой конкретную задачу и добиваться её выполнения.

*Воспитательные:*

- воспитывать устойчивый интерес к программированию;
- воспитывать потребность в самостоятельном изучении новых технологий;
- воспитывать информационную культуру как составляющую общей культуры современного человека;
- формировать потребность в творческой деятельности, стремление к самовыражению через техническое творчество;
- формировать новаторское отношение ко всем сферам жизнедеятельности человека.

### **Планируемые образовательные результаты**

В результате изучения курса обучающийся должен:

***Знать:***

- терминологию, применяемую в области сетевых технологий;
- основные типы сетевых протоколов передачи данных;
- принципы проектирования распределенных сетевых приложений;
- принципы структурирования данных;
- основные принципы использования баз данных;
- основные принципы организации очередей;
- основные принципы балансировки нагрузки между компонентами приложения.

***Уметь:***

- планировать архитектуру микросервисного приложения;
- программировать на языке Go;
- проектировать структуру базы данных;
- проектировать систему кэширования;
- проектировать систему очередей и взаимодействия компонентов приложения;
- пользоваться инструментами разработки, отладки и тестирования приложения;
- читать техническую документацию;
- анализировать, оформлять и представлять результаты работы.

***Владеть:***

- навыками планирования и настройки сетевой подсистемы Linux;
- навыками оформления технической документации программного кода;
- навыками практического проектирования и реализации микросервисных приложений.

### **Личностные результаты**

- готовность и способность к образованию, в том числе самообразованию, на протяжении всей жизни; сознательное отношение к непрерывному образованию как условию успешной профессиональной и общественной деятельности;

- навыки сотрудничества со сверстниками, взрослыми в образовательной, учебно-исследовательской, проектной и других видах деятельности;
- осознанный выбор будущей профессии и возможностей реализации собственных жизненных планов; отношение к профессиональной деятельности как возможности участия в решении личных, общественных, государственных, общенациональных проблем.

### Метапредметные результаты

1) умение самостоятельно определять цели деятельности и составлять планы деятельности; самостоятельно осуществлять, контролировать и корректировать деятельность; использовать все возможные ресурсы для достижения поставленных целей и реализации планов деятельности; выбирать успешные стратегии в различных ситуациях;

2) умение продуктивно общаться и взаимодействовать в процессе совместной деятельности, учитывать позиции других участников деятельности, эффективно разрешать конфликты;

3) владение навыками познавательной, учебно-исследовательской и проектной деятельности, навыками разрешения проблем; способность и готовность к самостоятельному поиску методов решения практических задач, применению различных методов познания;

4) готовность и способность к самостоятельной информационно-познавательной деятельности, включая умение ориентироваться в различных источниках информации, критически оценивать и интерпретировать информацию, получаемую из различных источников.

При разработке учебного курса были использованы следующие принципы:

- принцип последовательности в изучении понятийного аппарата в областях программирование и сетевые технологии;
- принцип системности знаний;
- принцип дифференцированности обучения;
- принцип фундаментальности знаний и умений;
- принцип доступности содержания курса;
- принцип связи теоретических знаний с практикой;
- принцип единой содержательной и процессуальной стороны обучения;
- принцип структурного единства содержания образования на разных уровнях его формирования с учетом личностного развития и становления школьника.

### Содержание курса

#### Раздел 1. Основы сетевой инфраструктуры – 2 часа

##### *Тема 1.1. Вводное занятие. Архитектура сетей передачи данных*

*Теория.* Основные сведения о сетевой инфраструктуре: сетевые узлы и интерфейсы, адресация, маршрутизация, система доменных имен DNS.

*Практика.* Знакомство с ОС Linux. Настройка сетевой подсистемы. Основные команды и утилиты для работы с сетью.

## Раздел 2. Изучение языка программирования Go – 20 часов

### ***Тема 2.1. Введение в программирование на языке Go***

***Теория.*** История создания и развития языка Go. Основные архитектурные особенности языка Go. Обоснование применения именно этого языка для разработки сетевых приложений.

***Практика.*** Установка и настройка программных средств для разработки на языке Go. Создание и компиляция первого приложения.

### ***Тема 2.2. Работа с переменными. Основные арифметические и логические операции***

***Теория.*** Типы переменных. Способы определения переменных. Использование указателей. Правила записи арифметических и логических выражений.

***Практика.*** Написание программного кода для закрепления изученной темы.

### ***Тема 2.3. Управление потоком выполнения программы. Циклы. Условия***

***Теория.*** Использование конструкций ветвления. Использование конструкций цикла. Синтаксис записи управляющих конструкций в языке Go.

***Практика.*** Написание программного кода для закрепления изученной темы.

### ***Тема 2.4. Упорядоченные структуры: массивы, срезы, карты***

***Теория.*** Основные операции с упорядоченными структурами: создание, добавление и удаление данных. Сортировка массивов. Объединение массивов. Вырезка из массивов.

***Практика.*** Написание программного кода для закрепления изученной темы.

### ***Тема 2.5. Функции. Обработка ошибок***

***Теория.*** Оформление кода в виде функций. Синтаксис определения функций. Обработка результатов и ошибок выполнения функций.

***Практика.*** Написание программного кода для закрепления изученной темы.

### ***Тема 2.6. Структуры и интерфейсы***

***Теория.*** Синтаксис определения структур. Понятие интерфейса. Структуры, как замена ООП.

***Практика.*** Написание программного кода для закрепления изученной темы.

### ***Тема 2.7. Многопоточность. Горутины, каналы***

***Теория.*** Реализация многопоточности в языке Go. Выполнение функций в разных потоках. Работа с каналами и организация передачи данных в многопоточном приложении.

***Практика.*** Написание программного кода для закрепления изученной темы.

### ***Тема 2.8. Пакеты (модули). Структурирование приложения***

***Теория.*** Разделение кода приложения на отдельные модули. Пакеты функций. Повторное использование кода.

***Практика.*** Написание программного кода для закрепления изученной темы.

### ***Тема 2.9. Стандартная библиотека Go***

***Теория.*** Обработка строковых переменных. Функции ввода/вывода. Работа с файловой системой.

*Практика.* Написание программного кода для закрепления изученной темы.

### ***Тема 2.10. Библиотека net для разработки сетевых приложений***

*Теория.* Основные функции для создания сетевых приложений. Работа с сокетами и сетевыми интерфейсами.

*Практика.* Написание программного кода для закрепления изученной темы.

## **Раздел 3. Протоколы передачи данных – 6 часов**

### ***Тема 3.1. Текстовые протоколы. HTTP***

*Теория.* Понятие сетевого протокола. Простые текстовые протоколы. Стандартный протокол HTTP. Средства языка Go для работы с HTTP.

*Практика.* Написание программного кода для закрепления изученной темы.

### ***Тема 3.2. Бинарные протоколы. gRPC***

*Теория.* Бинарные протоколы передачи данных. Проектирование и реализация бинарного протокола на примере gRPC.

*Практика.* Написание программного кода для закрепления изученной темы.

### ***Тема 3.2. Проектирование собственного протокола***

*Теория.* Основные принципы проектирования собственных протоколов передачи данных.

*Практика.* Написание программного кода для закрепления изученной темы.

## **Раздел 4. Работа с базами данных – 18 часов**

### ***Тема 4.1. Реляционные базы данных (SQL). Основы проектирования структуры БД, Средства языка Go для работы с SQL***

*Теория.* Основы структурированного представления данных. Принципы проектирования реляционных баз данных. Знакомство с функциями библиотеки SQLx для языка Go.

*Практика.* Установка и настройка сервера MySQL. Проектирование структуры базы данных.

### ***Тема 4.2. Основные операции SQL: выборка, вставка, изменение, удаление***

*Теория.* Работа с операторами языка SQL: SELECT, INSERT, UPDATE, DELETE.

*Практика.* Написание программного кода для закрепления изученной темы.

### ***Тема 4.3. Поиск и выборка данных***

*Теория.* Проектирование запросов SELECT с использованием различных условий для выборки данных.

*Практика.* Написание программного кода для закрепления изученной темы.

### ***Тема 4.4. Связи и отношения между таблицами. Сложные запросы***

**Теория.** Типы связей между объектами в реляционных БД. Таблицы «сущностей» и таблицы «связей». Проектирование сложных запросов на выборку данных из нескольких таблиц.

**Практика.** Написание программного кода для закрепления изученной темы.

#### **Тема 4.5. Паттерны проектирования приложений, использующих БД**

**Теория.** Основы проектирования приложений, использующих базы данных. Обработка ошибок и исключительных ситуаций при работе с сервером БД.

**Практика.** Проектирование архитектуры приложения для работы с БД.

#### **Тема 4.6. БД Redis. Кэширование данных**

**Теория.** Основные принципы использования кэширования в микросервисных приложениях. Средства языка Go, для работы с сервером Redis&

**Практика.** Проектирование системы кэширования. Написание программного кода для закрепления изученной темы.

#### **Тема 4.7. Нереляционные базы данных (NoSQL). БД MongoDB. Средства языка Go для работы с MongoDB**

**Теория.** Отличия реляционных и нереляционных баз данных. Основы проектирования БД MongoDB. Средства языка Go Для работы с MongoDB.

**Практика.** Установка и настройка сервера MongoDB. Написание программного кода для закрепления изученной темы.

#### **Тема 4.8. Основные операции MongoDB: выборка, вставка, изменение, удаление**

**Теория.** Изучение основных операций MongoDB. Построение сложных запросов на выборку данных.

**Практика.** Написание программного кода для закрепления изученной темы.

### **Раздел 5. Архитектура микросервисных приложений – 6 часов**

#### **Тема 5.1. Основы проектирования архитектуры микросервисных приложений**

**Теория.** Принципы проектирования архитектуры микросервисных приложений. Разбиение приложения на компоненты. Логирование. Метрики приложения.

**Практика.** Выполнение теста, направленного на закрепление темы.

#### **Тема 5.2. Сервер NUTS. Использование очередей для обмена данными между компонентами Go-приложения**

**Теория.** Основные принципы использования очередей сообщений и передачи сигналов между компонентами приложения. Средства языка Go для работы с сервером очередей.

**Практика.** Установка и настройка сервера NUTS. Написание программного кода для закрепления изученной темы.

### **Раздел 6. Балансировка нагрузки – 4 часа**

#### **Тема 6.1. Измерение и планирование распределения нагрузки приложения**

*Теория.* Способы измерения нагрузки на приложение. Утилиты для организации нагрузочного тестирования.

*Практика.* Написание программного кода для закрепления изученной темы.

### **Тема 6.2. Использование серверов Nginx и HA Proxy для балансировки нагрузки**

*Теория.* Основные способы распределения и балансировки нагрузки между компонентами микросервисного приложения.

*Практика.* Установка и настройка сервера Nginx. Конфигурация сервера для распределения нагрузки.

## **Раздел 7. Разработка индивидуального проекта – 10 часов**

### **Тема 7.1. Выбор темы. Планирование архитектуры приложения**

*Теория.* О выборе темы проекта. Поиск актуальных проблем и выбор путей их решения.

*Практика.* Самостоятельные изыскания для выбора темы проекта.

### **Тема 7.2. Разработка структуры БД**

*Практика.* Разработка структуры базы данных. Написание программного кода.

### **Тема 7.3. Реализация программного кода приложения**

*Практика.* Написание программного кода приложения.

### **Тема 7.4. Настройка системы балансировки нагрузки**

*Практика.* Проектирование системы балансировки нагрузки. Настройка сервера.

## **Раздел 8. Итоговое занятие. Защита проектов – 2 часа**

### **Тема 8.1. Выбор и обоснование темы**

*Практика.* Выступление с докладом и презентацией перед аудиторией, демонстрация работы проекта, ответы на вопросы.

### **Тематический план**

№	Названия раздела/темы	Кол-во часов			Формы контроля
		Всего	Теория	Практика	
1.	<b>Вводное занятие.</b>	2	1	1	
1.1	Архитектура сетей передачи данных	2	1	1	Текущий контроль. Практическое задание
2.	<b>Изучение языка программирования Go</b>	20	10	10	
2.1	Введение в программирование на языке Go	2	1	1	Текущий контроль. Практическое задание
2.2	Работа с переменными. Основные арифметические	2	1	1	Текущий контроль. Практическое задание



	и логические операции				
2.3	Управление потоком выполнения программы. Циклы. Условия	2	1	1	Текущий контроль. Практическое задание
2.4	Упорядоченные структуры: массивы, срезы, карты	2	1	1	Текущий контроль. Практическое задание
2.5	Функции. Обработка ошибок	2	1	1	Текущий контроль. Практическое задание
2.6	Структуры и интерфейсы	2	1	1	Текущий контроль. Практическое задание
2.7	Многопоточность. Горутины, каналы	2	1	1	Текущий контроль. Практическое задание
2.8	Пакеты (модули). Структурирование приложения	2	1	1	Текущий контроль. Практическое задание
2.9	Стандартная библиотека Go	2	1	1	Текущий контроль. Практическое задание
2.10	Библиотека net для разработки сетевых приложений	2	1	1	Текущий контроль. Промежуточная аттестация
<b>3.</b>	<b>Протоколы передачи данных</b>	<b>6</b>	<b>3</b>	<b>3</b>	
3.1	Текстовые протоколы. HTTP	2	1	1	Текущий контроль. Практическое задание
3.2	Бинарные протоколы. gRPC	2	1	1	Текущий контроль. Практическое задание
3.3	Проектирование собственного протокола	2	1	1	Текущий контроль. Практическое задание
<b>4.</b>	<b>Работа с базами данных</b>	<b>18</b>	<b>10</b>	<b>8</b>	
4.1	Реляционные базы данных (SQL). Основы проектирования структуры БД, Средства языка Go для работы с SQL	2	2	0	Текущий контроль. Тест
4.2	Основные операции SQL: выборка, вставка, изменение, удаление	2	1	1	Текущий контроль. Практическое задание
4.3	Поиск и выборка данных	2	1	1	Текущий контроль. Практическое задание
4.4	Связи и отношения между таблицами. Сложные запросы	2	1	1	Текущий контроль. Практическое задание
4.5	Паттерны проектирования приложений, использующих БД	2	1	1	Текущий контроль. Практическое задание
4.6	БД Redis. Кэширование данных	4	2	2	Текущий контроль. Практическое задание
4.7	Нереляционные базы	2	1	1	Текущий контроль.

	данных (NoSQL). БД MongoDB. Средства языка Go для работы с MongoDB				Практическое задание
4.8	Основные операции MongoDB: выборка, вставка, изменение, удаление	2	1	1	Текущий контроль. Промежуточная аттестация
<b>5.</b>	<b>Архитектура микросервисных приложений</b>	<b>6</b>	<b>4</b>	<b>2</b>	
5.1	Основы проектирования архитектуры микросервисных приложений	2	2	0	Текущий контроль. Тест
5.2	Сервер NUTS. Использование очередей для обмена данными между компонентами Go-приложения	4	2	2	Текущий контроль. Практическое задание
<b>6.</b>	<b>Балансировка нагрузки</b>	<b>4</b>	<b>2</b>	<b>2</b>	
6.1	Измерение и планирование распределения нагрузки приложения	2	1	1	Текущий контроль. Тест
6.2	Использование серверов Nginx и HA Proxy для балансировки нагрузки	2	1	1	Текущий контроль. Практическое задание
<b>7.</b>	<b>Разработка индивидуального проекта</b>	<b>10</b>	<b>2</b>	<b>8</b>	
7.1	Выбор темы. Планирование архитектуры приложения	2	1	1	Текущий контроль. Практическое задание
7.2	Разработка структуры БД	1	0	2	Текущий контроль. Практическое задание
7.3	Реализация программного кода приложения	5	1	4	Текущий контроль. Практическое задание
7.4	Настройка системы балансировки нагрузки	1	0	1	Текущий контроль. Практическое задание
<b>8.</b>	<b>Итоговое занятие. Защита проектов</b>	<b>2</b>	<b>0</b>	<b>2</b>	
8.1	Защита проекта и анализ результатов работы	2	0	2	Итоговая аттестация. Защита проектов
<b>Всего часов:</b>		<b>68</b>			

## ФОРМЫ РАБОТЫ

Теоретические занятия предназначены для представления теоретических знаний по учебному курсу. Проводятся в виде лекций.

В данном элективном курсе предполагается два вида практических занятий: работа с учебным оборудованием и программирование. Все темы включают в себя самостоятельную работу. Самостоятельная работа предусматривает несколько видов деятельности ученика: работу с литературой, работу на компьютере, выполнение практических заданий.

## **ФОРМЫ КОНТРОЛЯ И ОЦЕНОЧНЫЕ МАТЕРИАЛЫ**

Формы контроля и оценочные материалы служат для определения результативности освоения Программы обучающимися. Текущий контроль проводится по окончании изучения каждой темы – выполнение обучающимися практических заданий. Промежуточный контроль проходит в середине учебного года в форме открытого занятия.

Итоговый контроль (зачетное занятие) проходит в конце учебного года – в форме конференции.

Формы проведения аттестации:

- практическое задание;
- самостоятельная работа;
- тест;
- конференция.

## **Список литературы**

1. Алан А.А. Донован, Брайан У.Керниган. Язык программирования Go. Вильямс, 2016, 432 стр.
2. Марк Саммерфильд. Программирование на Go. Разработка приложений XXI века. ДМК Пресс, 2016, 580 стр.
3. Алан Бьюли. Изучаем SQL. Символ-Плюс, 2007, 309 стр.
4. Бэнкер Кайл. MongoDB в действии. ДМК Пресс, 2017, 394 стр.
5. Ричардсон Крис. Микросервисы. Паттерны разработки и рефакторинга. Питер, 2019, 544 стр.

## **Интернет-ресурсы**

1. Введение в программирование на Go. <http://golang-book.ru>
2. Эффективный Go. [https://github.com/Konstantin8105/Effective\\_Go\\_RU](https://github.com/Konstantin8105/Effective_Go_RU)
3. Разработка WEB-приложений на языке Go. <https://github.com/astaxie/build-web-application-with-golang/blob/master/ru/preface.md>
4. Карл Сегуин. Маленькая книга о Redis. <https://thesunwave.gitbooks.io/redis/content/>