

Современные ВЕБ-технологии

02.03.03 -Математическое обеспечение и администрирование информационных систем,
направленность (профиль) -разработка и администрирование информационных систем

<http://vikchas.ru>

Лекция 1. Тема «Кодирование на языках программирования и сопутствующие технологии»

Часовских Виктор Петрович
доктор технических наук, профессор кафедры
ШИиКМ, ФГБОУ ВО «Уральский государственный
экономический университет

Екатеринбург 2025

Кодирование на языках программирования и сопутствующие технологии

1. Кодирование на языках программирования

Языки программирования и их парадигмы

Императивное программирование:

Пример на Python

```
def calculate_sum(numbers):  
    sum = 0  
    for number in numbers:  
        sum += number  
    return sum
```

// Пример на C#

```
public int CalculateSum(int[] numbers)  
{  
    int sum = 0;  
    foreach(int number in numbers)  
    {  
        sum += number;  
    }  
    return sum;  
}
```

Объектно-ориентированное программирование:

Пример на Python

```
class Person:  
    def __init__(self, name, age):
```

```
self.name = name
```

```
self.age = age
```

```
def greet(self):
```

```
    return f"Привет, меня зовут {self.name}, мне {self.age} лет."
```

// Пример на C#

```
public class Person
```

```
{
```

```
    public string Name { get; set; }
```

```
    public int Age { get; set; }
```

```
    public Person(string name, int age)
```

```
{
```

```
        Name = name;
```

```
        Age = age;
```

```
}
```

```
    public string Greet()
```

```
{
```

```
        return $"Привет, меня зовут {Name}, мне {Age} лет.";
```

```
}
```

```
}
```

Функциональное программирование:

Пример на Python

```
numbers = [1, 2, 3, 4, 5]
```

```
doubled = list(map(lambda x: x * 2, numbers))
```

```
filtered = list(filter(lambda x: x > 5, doubled))
```

// Пример на C#

```
var numbers = new[] { 1, 2, 3, 4, 5 };  
var doubled = numbers.Select(x => x * 2);  
var filtered = doubled.Where(x => x > 5);
```

2. Предметная область автоматизации

Автоматизация включает применение технологий для выполнения задач с минимальным участием человека:

Типы автоматизации:

- **Бизнес-процессы:** Системы учета, CRM, ERP, документооборот
- **Производственные процессы:** АСУТП, системы контроля качества
- **Тестирование ПО:** Автоматизация проверок с использованием Selenium, JUnit, PyTest

Пример автоматизации бизнес-процесса (обработка заказов):

```
def process_order(order):  
    if validate_order(order):  
        reserve_inventory(order.items)  
        charge_customer(order.customer, order.total)  
        schedule_shipping(order)  
        notify_customer(order, "Ваш заказ принят в обработку")  
        return True  
    else:  
        notify_customer(order, "Проблема с заказом. Пожалуйста, свяжитесь с поддержкой")  
        return False
```

3. Основы современных систем управления базами данных

Реляционные СУБД (MySQL, PostgreSQL, MS SQL Server, Oracle)

Создание таблицы:

SQL

```
CREATE TABLE Customers (  
    CustomerID INT PRIMARY KEY AUTO_INCREMENT,  
    Name VARCHAR(100) NOT NULL,  
    Email VARCHAR(100) UNIQUE,  
    RegistrationDate DATETIME DEFAULT CURRENT_TIMESTAMP  
);
```

Основные операции CRUD:

SQL

-- Create (Создание)

```
INSERT INTO Customers (Name, Email) VALUES ('Иван Иванов',  
'ivan@example.com');
```

-- Read (Чтение)

```
SELECT * FROM Customers WHERE CustomerID = 1;
```

-- Update (Обновление)

```
UPDATE Customers SET Email = 'new.ivan@example.com' WHERE  
CustomerID = 1;
```

-- Delete (Удаление)

```
DELETE FROM Customers WHERE CustomerID = 1;
```

NoSQL СУБД (MongoDB, Cassandra, Redis)

Пример работы с MongoDB:

JAVASCRIPT

// Создание документа

```
db.customers.insertOne({  
    name: "Иван Иванов",
```

```

    email: "ivan@example.com",
    registrationDate: new Date()
  });

// Чтение документа
db.customers.findOne({ email: "ivan@example.com" });

// Обновление документа
db.customers.updateOne(
  { email: "ivan@example.com" },
  { $set: { email: "new.ivan@example.com" } }
);

// Удаление документа
db.customers.deleteOne({ email: "new.ivan@example.com" });

```

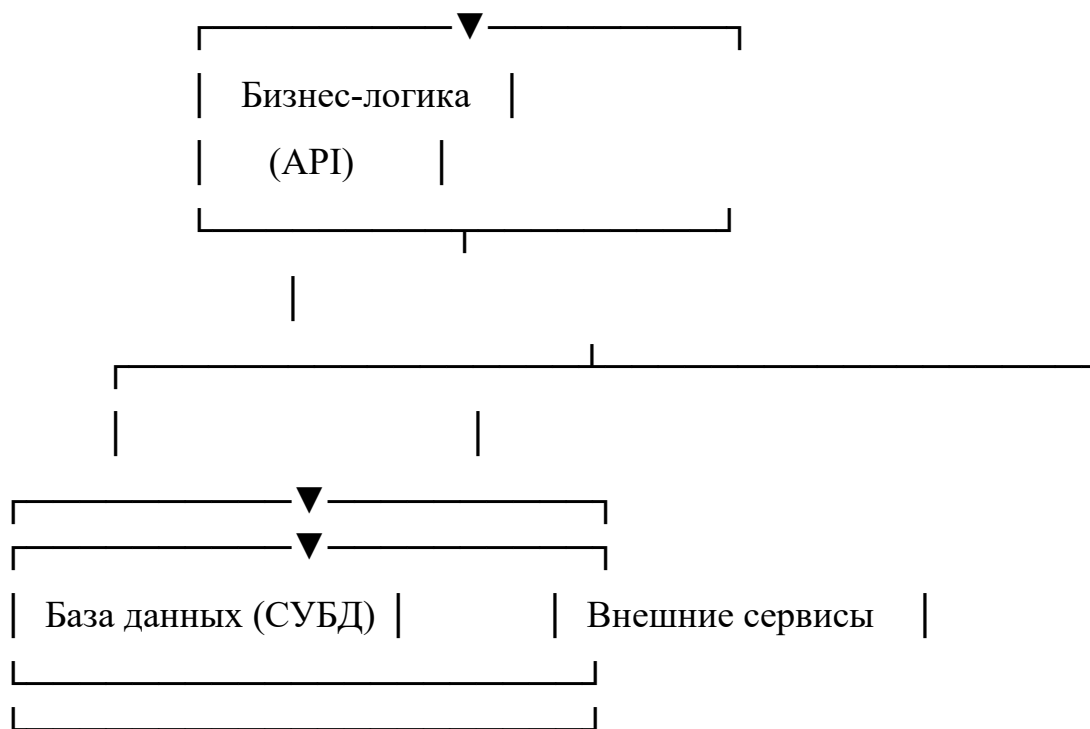
4. Устройство и функционирование современных информационных систем

Архитектуры ИС:

- **Монолитная архитектура**- единое приложение, включающее все компоненты
- **Микросервисная архитектура**- набор небольших специализированных сервисов
- **Клиент-серверная архитектура**- клиентская и серверная части работают отдельно

Компоненты типичной ИС:





5. Теория баз данных

Нормализация данных

Пример ненормализованной таблицы:

OrderID | Customer | Products

OrderID	Customer	Products
1	Иванов	Ноутбук (2 шт), Мышь (1 шт)
2	Петров	Клавиатура (1 шт), Монитор (1 шт)

После нормализации:

Customers:

CustomerID | Name

CustomerID	Name
1	Иванов
2	Петров

Orders:

OrderID | CustomerID

OrderID	CustomerID
1	1
2	2

1	1
2	2

Products:

ProductID	ProductName	Price
-----------	-------------	-------

-----	-----	-----
-------	-------	-------

1	Ноутбук	50000
2	Мышь	1000
3	Клавиатура	2000
4	Монитор	15000

OrderDetails:

OrderID	ProductID	Quantity
---------	-----------	----------

-----	-----	-----
-------	-------	-------

1	1	2
1	2	1
2	3	1
2	4	1

ER-диаграммы

Визуальное представление структуры базы данных, показывающее сущности и их взаимосвязи.

Типы связей:

- Один к одному (1:1)
- Один ко многим (1:N)
- Многие ко многим (M:N)

6. Системы хранения и анализа баз данных

Хранилища данных (Data Warehousing)

OLTP vs OLAP:

- OLTP (Online Transaction Processing)- для обработки транзакций

- **OLAP (Online Analytical Processing)**- для аналитики и отчётности

ETL-процессы (Extract, Transform, Load)

PYTHON

Пример ETL-процесса на Python

```
def etl_process():
```

```
    # Extract - извлечение данных
```

```
    raw_data = extract_from_source()
```

```
    # Transform - преобразование данных
```

```
    transformed_data = transform_data(raw_data)
```

```
    # Load - загрузка данных в хранилище
```

```
    load_to_warehouse(transformed_data)
```

Системы бизнес-аналитики (BI)

- Инструменты: Power BI, Tableau, QlikView
- Предоставляют визуализацию и анализ больших объемов данных для принятия бизнес-решений

Примеры аналитических запросов:

SQL

-- Анализ продаж по категориям

```
SELECT Category, SUM(Sales) as TotalSales
```

```
FROM Sales
```

```
GROUP BY Category
```

```
ORDER BY TotalSales DESC;
```

-- Сравнение продаж по годам

```
SELECT YEAR(SaleDate) as Year, SUM(Amount) as TotalSales
```

```
FROM Sales
```

```
GROUP BY YEAR(SaleDate)
```

ORDER BY Year;

Большие данные (Big Data)

- Технологии: Hadoop, Spark, Kafka
- Позволяют обрабатывать огромные объемы данных, которые не помещаются в традиционных СУБД.

PYTHON

Пример обработки данных с Apache Spark

```
from pyspark import SparkContext
```

```
from pyspark.sql import SparkSession
```

```
spark = SparkSession.builder.appName("DataAnalysis").getOrCreate()
```

Чтение данных

```
data = spark.read.csv("hdfs://bigdata/sales.csv", header=True)
```

Агрегация и анализ

```
result = data.groupBy("category").sum("sales_amount")
```

Сохранение результатов

```
result.write.parquet("hdfs://bigdata/results/")
```

Данные технологии и концепции составляют основу современной разработки информационных систем, обеспечивая эффективную работу с данными и автоматизацию бизнес-процессов.