

# Разработчик Java

Курс о языке Java, особенностях платформы, стандартной библиотеке, о проектировании и тестировании, о том, как работать с базами, файлами, веб-фронтендом и другими приложениями.

Начало занятий

В марте

## 1 Язык и платформа Java

### 1 1.1 Подготовка к курсу

Введение. Обзор и задачи курса. Настройка окружения: система контроля версий git, система контроля зависимостей maven, среда разработки IntelliJ IDEA. Сборка и запуск java приложений.

Домашние задания

#### 1 ДЗ 01. Сборка и запуск проекта

Создать проект под управлением maven в IntelliJ IDEA.

Добавить зависимость на Google Guava/Apache Commons/библиотеку на ваш выбор.  
Использовать библиотечные классы для обработки входных данных.

Задать имя проекта (project\_name) в pom.xml

Собрать project\_name.jar содержащий все зависимости.

Проверить, что приложение можно запустить из командной строки.

Выложить проект на github.

Создать ветку "obfuscation" изменить в ней pom.xml, так чтобы сборка содержала стадию обфускации байткода.

---

## 2 1.2 Байт код

Байт код. Содержание .class. Декомпиляция. Обфускация. История изменений языка Java от версии к версии. Разбор ДЗ.

---

## 3 2.1 Углубленные основы

Углубленные основы. Загрузка классов. Виды и задачи class loader-ов. Класс Class<?>. Примитивные типы, строки, массивы. Память, которую занимают объекты.

Домашние задания

### 1 ДЗ 02. Измерение памяти

Напишите стенд для определения размера объекта.

Передавайте для измерения в стенд фабрику объектов.

Определите размер пустой строки и пустых контейнеров. Определите рост размера контейнера от количества элементов в нем.

Если вы знакомы с инструментированием, сравните результаты измерений.

---

## 4 2.2 Remote debug

Hot swap. Remote debug. Сборщик мусора. Instrumentation. Разбор примеров и ДЗ.

---

## 5 3.1 Контейнеры и алгоритмы

Generics. Контейнеры и алгоритмы. Обзор устройства и работы контейнеров из java.util. Сравнение контейнеров. Карты на основе хэш функции и на основе дерева. Разбор алгоритмов из java.util.Collections.

Домашние задания

### 1 ДЗ 03. MyArrayList

Написать свою реализацию ArrayList на основе массива.

```
class MyArrayList<T> implements List<T>{...}
```

Проверить, что на ней работают методы  
addAll(Collection<? super T> c, T... elements)  
static <T> void copy(List<? super T> dest, List<? extends T> src)  
static <T> void sort(List<T> list, Comparator<? super T> c)

из java.util.Collections

---

## 6 3.2 Инструменты для преобразования контейнеров

Инструменты для преобразования контейнеров. Apache Commons. Google Guava. Разбор примеров и ДЗ.

---

## 7 4.1 Сборщик мусора

Параметры запуска VM. Сборщик мусора. Виды сборок. Разделение памяти под разные поколения объектов. Виды сборщиков. JMX. Управление приложением из jconsole. OutOfMemory. dump памяти. Исследование thread dump и heap dump.

Домашние задания

### 1 ДЗ 04. Измерение активности GC

Написать приложение, которое следит за сборками мусора и пишет в лог количество сборок каждого типа (young, old) и время которое ушло на сборки в минуту.

Добиться OutOfMemory в этом приложении через медленное подтекание по памяти (например добавлять элементы в List и удалять только половину).

Настроить приложение (можно добавлять Thread.sleep(...)) так чтобы оно падало с ООМ примерно через 5 минут после начала работы. Собрать статистику (количество сборок, время на сборки) по разным типам GC. Сделать выводы.

---

## 8 4.2 QA и тестирование

QA, тестирование. Виды тестов. "Заглушки". Testing frameworks: junit, mockito. Разбор примеров и ДЗ.

---

9	<b>5.1 Аннотации</b>	Аннотации. Стандартные аннотации. Применение аннотаций. Типы аннотаций. Синтаксис. Создание своих аннотаций. Reflection.
		Домашние задания
		1 ДЗ 05. Тестовый фреймворк на аннотациях
		Написать свой тестовый фреймворк. Поддержать свои аннотации @Test, @Before, @After. Запускать вызовом статического метода с именем класса с тестами.
10	<b>5.2 Java 8</b>	Элементы функционального программирования. Streams. Lambdas. Примеры.
11	<b>6.1 Типы ссылок. Кэширование</b>	Типы ссылок в Java. Системы кэширования. Ehcache.
		Домашние задания
		1 ДЗ-06: my cache engine
		Напишите свой cache engine с soft references.
12	<b>6.2 Логгирование</b>	java.util.logging, log4j, slf4j. Разбор примеров и ДЗ.

### 1 7.1 Концепты проектирования ООП

Понятие дизайна приложения и паттерна проектирования. Концепты проектирования ООП: delegation, aggregation, composition, association, coupling and cohesion. SOLID.

Домашние задания

#### 1 ДЗ 06. Написать эмулятор АТМ

Написать эмулятор АТМ (банкомата).

Объект класса АТМ должен уметь

- принимать банкноты разных номиналов (на каждый номинал должна быть своя ячейка)
- выдавать запрошенную сумму минимальным количеством банкнот или ошибку если сумму нельзя выдать
- выдавать сумму остатка денежных средств

### 2 7.2 Behavioral patterns

Behavioral patterns часть 1. Разбор примеров и ДЗ.

### 3 8.1 Structural patterns

Behavioral patterns часть 2. Structural patterns.

Домашние задания

#### 1 ДЗ 07. ATM Department

Написать приложение ATM Department:

- Приложение может содержать несколько ATM
- Department может собирать сумму остатков со всех ATM
- Department может инициировать событие – восстановить состояние всех ATM до начального.

(начальные состояния у разных ATM могут быть разными)

---

### 4 8.2 Creational patterns

Creational patterns. Связь между паттернами, "похожие" паттерны. Антипаттерны. Разбор примеров и ДЗ.

- 1 **9.1  
Сериализация** Java IO. Сериализация. Запись и чтение бинарных файлов. Работа с simple-json, javax.json и com.google.code.gson.
- Домашние задания
- 1 ДЗ-08: JSON object writer
- Напишите свой json object writer (object to JSON string) аналогичный gson на основе javax.json или simple-json и Reflection.  
Поддержите массивы объектов и примитивных типов, и коллекции из стандартной библиотеки.
- 
- 2 **9.2 XML** XML. Java NIO. Разбор примеров и ДЗ.
- 
- 3 **10.1 JDBC** Работа с реляционной базой. JDBC. ORM через паттерн Executor. JPA.
- Домашние задания



## 1 ДЗ-10: myORM

Создайте в базе таблицу с полями:

- id bigint(20) NOT NULL auto\_increment
- name varchar(255)
- age int(3)

Создайте абстрактный класс DataSet. Поместите long id в DataSet.

Добавьте класс UserDataSet (с полями, которые соответствуют таблице) унаследуйте его от DataSet.

Напишите Executor, который сохраняет наследников DataSet в базу и читает их из базы по id и классу.

```
<T extends DataSet> void save(T user){...}  
<T extends DataSet> T load(long id, Class<T>  
clazz){...}
```

Проверьте его работу на UserDataSet

---

## 4 10.2 myBatis

Inmemory базы. myBatis. Разбор примеров и ДЗ.

---

## 5 11.1 Hibernate

Hibernate. Конфигурирование Hibernate в коде и в XML. Java persistency query language. DBService pattern.

Домашние задания

### 1 ДЗ-11: Hibernate ORM

На основе ДЗ 10:

1. Оформить решение в виде DBService (interface DBService, class DBServiceImpl, UsersDAO, UsersDataSet, Executor)

2. Не меняя интерфейс DBService сделать DBServiceHibernateImpl на Hibernate.

3. Добавить в UsersDataSet поля:

адресс (OneToOne)

```
class AddressDataSet{  
private String street;  
}
```

и телефон\* (OneToMany)

```
class PhoneDataSet{  
private String number;  
}
```

Добавить соответствующие датасеты и DAO.

\* можно не поддерживать в ДЗ 10

---

## 6 11.2 JPQL

Connection Pools. JPQL. Разбор примеров и ДЗ.

---

7 **12.1 Web сервер**

Встроенный веб сервер. Сервлеты: servlet API, жизненный цикл сервлета. Jetty: устройство, работа, подключение сервлетов.

Домашние задания

1 ДЗ-12: Веб сервер

Встроить веб сервер в приложение из ДЗ-11.  
Сделать админскую страницу, на которой можно добавить пользователя, получить имя пользователя по id и получить количество пользователей в базе.

---

8 **12.3 No SQL**

No SQL. Работа с MongoDB. Разбор примеров и ДЗ.

**1 13.1 Thread**

Многопоточность. Класс и объект Thread. Создание многопоточного приложения. Проблемы многопоточного доступа.

Домашние задания

**1 ДЗ-14: Многопоточная сортировка**

Написать приложение, которое сортирует массив чисел в 4 потоках с использованием библиотеки или без нее.

---

**2 13.2 JMM**

Happens before. Cooperation. Atomics.

---

**3 14.1 Dependency injection**

Web archive. Jetty as an external web server. Spring IoC. Spring beans.

Домашние задания

**1 ДЗ-13: WAR**

Собрать war для приложения из ДЗ-12.  
Создавать кэш и DBService как Spring beans, передавать (inject) их в сервлеты.  
Запустить веб приложение во внешнем веб сервере.

---

**4 14.2 Asynchronous Web applications**

AJAX. Long polling. Websockets. Разбор примеров из ДЗ.

---

## 5 15.1 Message System

Потокобезопасные контейнеры. Система обмена сообщениями.

Домашние задания

### 1 ДЗ-15: MessageSystem

Добавить систему обмена сообщениями в веб сервер из ДЗ-13.

Пересылать сообщения из вебсокета в DBService и обратно.

Организовать структуру пакетов без циклических зависимостей.

---

## 6 15.2 Executors

Поддержка многопоточности в стандартной библиотеке. Workers. Executors. Fork/Join. Разбор примеров и ДЗ.

---

7 **16.1**  
**Многопроцессные приложения**

Сокеты. RMI. Разбор примера многопроцессного приложения.

Домашние задания

1 ДЗ-16: MessageServer

Сервер из ДЗ-15 разделить на три приложения:

- MessageServer
- Frontend
- DBServer

Запускать Frontend и DBServer из MessageServer.

Сделать MessageServer сокет-сервером, Frontend и DBServer клиентами.

Пересылать сообщения с Frontend на DBService через MessageServer.

Запустить приложение с двумя фронтендами (на разных портах)\* и двумя датабазными серверами.

\* если у вас запуск веб приложения в контейнере, то MessageServer может копировать root.war в контейнеры при старте

---

8 **16.2 NIO**

Registration. NIO для сокетов. Проектная работа.

# 5 Проектная работа

1 17.1  
Консультация  
по ДЗ и  
проектам

Домашние задания

1 Проектная  
работа

---

2 18.1  
Консультация  
по ДЗ и  
проектам

---

3 Защита  
проектов