**Полезные ссылки от профессора-5**

[Java Syntax](https://javarush.ru/quests/QUEST_JAVA_SYNTAX)

[Уровень 5](https://javarush.ru/quests/lectures?quest=QUEST_JAVA_SYNTAX&level=5), Лекция 10

— Приветствую, Амиго! Ты, как я понял, пришёл на мою лекцию по физическим основам пространственно-временного искривления труб? Нет? А зачем тогда? Нужны дополнительные материалы по пятому уровню? Что ж, держи тогда.

**Зачем нужны конструкторы?**

Ты уже успел задать себе этот вопрос? А ответ нашёл? А ты уверен, что ответ — правильный? Давай проверим! Как создать дефолтного кота без признаков, а как создать такого же кота, но с определённым окрасом и тембром голоса? Не знаешь? Вот тебе [отличная статья](https://javarush.ru/groups/posts/konstruktory-v-java-1) по самым основам конструкторов в Java. Читай и просвещайся!

**Конструкторы базовых классов**

Ты тут начал разбираться с конструкторами в Java. Поэтому тебе не помешает одна [интересная статья](https://javarush.ru/groups/posts/1927-konstruktorih-bazovihkh-klassov--), которую я нашёл в закромах нашего корабля. Она посвящена конструкторам базовых классов, и как раз рассчитана на твой уровень. А ещё из неё ты узнаешь (или повторишь) о том, что такое суперкласс и класс-наследник, в каком порядке происходит вызов конструкторов и в каком порядке инициализируются поля.

Сегодня не очень много статей, зато тема очень важная. Так что углубись в неё. А если захочешь почитать что-то ещё по теме конструкторов — **Head First Java** тебе в помощь! Или, например «**Библиотека профессионала**» Кея Хорстманна — тоже отличный учебник. Думаешь, тебе ещё рано её читать, потому что ты — ещё не профессионал? А вот и зря. Профессионалами не рождаются, помнишь?

**Геттеры и сеттеры**

Какое-то время назад ты не знал, что такое инкапсуляция и зачем она нужна. А может и сейчас не чувствуешь себя уверенно, когда речь заходит о сокрытии данных классов и помощниках Java в этом деле — геттерах и сеттерах. Если это так, то будь добр, ознакомься с [очень полезной лекцией](https://javarush.ru/groups/posts/1928-getterih-i-setterih), которая, скорее всего, упрочнит твоё понимание инкапсуляции.

**Тернарный оператор**

Этот зверёк очень непривычный для новичка, и по большому счёту можно обойтись вообще без него… но ведь тернарный оператор так хорошо и просто сокращает код! Поэтому если ты пока не знаешь об этом заменителе конструкции if-else, рекомендую [ознакомиться](https://javarush.ru/groups/posts/ternarnyj-operator) и потихоньку начинать вплетать его в свой код.

А ещё я решил тебе порекомендовать книгу на будущее. Поскольку в лекции о тернарном операторе шла речь о читаемости кода, запомни это название — «**Чистый код**» и её автора — Роберта Мартина.

В этой книге собраны лучшие практики и рекомендации для программистов, которые позволят тебе писать не только работающий, но и легко читаемый код.