**Полезные ссылки от профессора — 12**

[Java Core](https://javarush.ru/quests/QUEST_JAVA_CORE)

[Уровень 2](https://javarush.ru/quests/lectures?quest=QUEST_JAVA_CORE&level=2), Лекция 10

— Ну здравствуй, Амиго! Хорошие новости — я заканчиваю работу над Самым-точным-предсказателем!

— Круто, а что он делает? Он скажет, когда я стану крутым программистом?

— Ну, не спеши, юный робот! Так далеко я не заглядывал. Но я уже знаю, что…

— Что?!

— … Ты пришел ко мне с вопросами по темам, которые изучил на этой неделе. Поэтому заранее подготовил для тебя дополнительные материалы: они помогут тебе разобраться.

**Практика использования полиморфизма**

Главное преимущество полиморфизма — гибкость. С одной стороны, ты можешь работать с несколькими типами данных так, будто это один тип. С другой стороны, благодаря этому принципу ты можешь сохранять поведение объектов. Когда тебе пригодится приведение к единому виду, а когда — уникальные характеристики? [Об этом и поговорим](https://javarush.ru/groups/posts/1974-praktika-ispoljhzovanija-polimorfizma).

**Как устроен механизм переопределения методов**

Ты уже знаком с перегрузкой методов. Пришло время [узнать о переопределении методов](https://javarush.ru/groups/posts/1975-kak-ustroen-mekhanizm-pereopredelenija-metodov-). Это поможет тебе в работе, когда будет нужно, чтобы общий метод выполнял разные действия в зависимости от класса, в котором его вызвали. Все возможно! Главное — знать как :)

**Для чего в Java нужны интерфейсы**

[В этой лекции](https://javarush.ru/groups/posts/1981-dlja-chego-v-java-nuzhnih-interfeysih) без спешки и подробно описано, что такое интерфейсы и почему они появились в языке. А еще ты узнаешь о популярных интерфейсах в Java. Приготовься, это тема с продолжением!

**Дефолтные методы в интерфейсах**

Каждая версия Java отличается от предыдущих. В восьмой появилось понятие дефолтных методов в интерфейсах. Благодаря этому ты можешь определять методы по умолчанию (default) и реализовывать их внутри интерфейса. Примеры и пояснения найдешь [в этой лекции](https://javarush.ru/groups/posts/1984-defoltnihe-metodih-v-interfeysakh).

**Абстрактные классы в Java на конкретных примерах**

Ты познакомился с понятием абстрактных классов и уже знаешь, что они служат своеобразными «заготовками» твоих будущих классов. Но обязательно ли все методы такого класса должны быть абстрактными? И все-таки, почему в Java нет множественного наследования классов? Вот тебе «наводка» от Самого-точного предсказателя: материал из [этой лекции](https://javarush.ru/groups/posts/1973-abstraktnihe-klassih-v-java-na-konkretnihkh-primerakh) лучше подготовит тебя к следующему уровню обучения.