**Сохранение данных/externalizable**

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

[Уровень 10](https://javarush.ru/quests/lectures?quest=QUEST_JAVA_CORE&level=10), Лекция 6

— Привет, Амиго! Хотел бы немного дополнить то, что рассказала тебе Элли.

Иногда бывает нужно управлять процессом сериализации. Вот одни из причин:

**1) Объект не готов к сериализации**: его нынешнее внутреннее состояние в процессе изменения.

**2) Объект содержит несериализуемые объекты, но может перевести их в удобную для сериализации форму:** сохранить в массив байт или что-нибудь еще.

**3) Объект хочет десериализовать все свои данные, как одно целое и/или зашифровать их перед сериализацией.**

Причин выполнить сериализацию в ручном режиме может быть множество. Но не хотелось бы лишаться всех тех преимуществ, которые дает стандартная сериализация. Ведь наш объект могут использовать другие объекты. И они тоже не смогут сериализоваться, если наш объект откажется от поддержки сериализации.

На этот случай тоже есть решение – интерфейс **Externalizable**. Спасибо дальновидным разработчикам Java. Достаточно заменить интерфейс Serializable на интерфейс Externalizable, и ваш класс сможет управлять процессом сериализации в ручном режиме.

Дело в том, что интерфейс **Externalizable**, в отличие от **Serializable**, содержит два метода, которые вызываются Java-машиной при сериализации объекта. Вот как это выглядит:

Код

class Cat implements Externalizable

{

public String name;

public int age;

public int weight;

public void writeExternal(ObjectOutput out)

{

out.writeObject(name);

out.writeInt(age);

out.writeInt(weight);

}

public void readExternal(ObjectInput in)

{

name = (String) in.readObject();

age = in.readInt();

weight = in.readInt();

}

}

Ничего не напоминает?

— Ух ты! Именно так мы пробовали сохранять объекты до «изобретения» сериализации.

— Да, теперь все просто: если устраивает стандартная сериализация – просто наследуем наш класс от интерфейса **Serializable**. Если не устраивает – наследуем от **Externalizable** и пишем код для сохранения/загрузки объекта нашего класса.

— А класс, помеченный **Externalizable**, считается сериализуемым? Мы можем «безопасно» хранить на него ссылки в наших сериализуемых классах?

— Да. Если класс реализует интерфейс **Serializable** или **Externalizable**, он считается сериализуемым.

— Отличное решение. Мне нравится.

— Рад это слышать. Но это еще не всё… Спроси лучше профессора Ханса о всяких нюансах. Они тут точно есть. Он хотел тебе дать что-то почитать.