**BufferedReader/InputStreamReader**

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

[Уровень 9](https://javarush.ru/quests/lectures?quest=QUEST_JAVA_CORE&level=9), Лекция 6

— Привет, Амиго! Ты уже давно пользуешься классами **BufferedReader** и **InputStreamReader**. Теперь давай разберемся, что же они делают.

Класс **InputStreamReader** является классическим адаптером от интерфейса **InputStream** к интерфейсу **Reader**. Тут даже и добавить нечего.

Но вкратце, происходит вот что. Когда ты запрашиваешь (читаешь посредством read) у объекта типа **InputStreamReader** очередной символ, он читает из переданного ему в конструктор **InputStream**-а несколько байт и возвращает их как один char.

Но Reader – не самый удобный объект для работы. Нам часто нужно прочитать не все введенные пользователем символы за один раз, а разбить эти символы на строки.

— Но у класса Reader ведь есть метод **read**(CharBuffer s), разве мы не можем использовать его?

— Этот метод читает данные куском, размер которого равен размеру буфера и помещает их в объект CharBuffer.

Текст обычно делится на строки (линии, line — строка по-английски). Поэтому метод read(CharBuffer s) может прочитать несколько «линий» за один раз. Если же нам нужно прочитать текст именно до конца «линии» (т.е. все символы строки до enter), для этого лучше поискать что-нибудь еще. И такой метод есть. В классе **BufferedReader**.

Класс **BufferedReader**, являющийся удобной надстройкой над Reader, имеет один очень удобный метод: **readLine()**. Этот метод позволяет читать из Reader-а сразу целыми строками (т.е. линиями). Когда ты в своем коде вызываешь метод readLine, он читает из объекта Reader символы, пока не встретится enter. Как только enter встретится, метод склеит символы в одну строку и вернет ее.

— Я этим постоянно пользовался, только не знал как все это работает. Теперь знаю. Спасибо, Ким.