**Пишем свой ридер-обертку на System.out**

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

[Уровень 9](https://javarush.ru/quests/lectures?quest=QUEST_JAVA_CORE&level=9), Лекция 7

— Привет, Амиго! Сегодня я расскажу как сделать новую интересную штуку — подменить объект System.out.  
**System.out** — это статическая переменная **out** типа **PrintStream** в классе **System**. Эта переменная имеет модификатор **final**, так что просто так новое значение ей не присвоить. Но класс System имеет для этого специальный метод **setOut(PrintStream stream)**. Им то мы и воспользуемся.

— Интересно. А на что мы его заменим?

— Нам нужен какой-то объект, куда можно будет собирать выведенные данные. Лучше всего на эту роль подойдет **ByteArrayOutputStream**. Это специальный класс, который с одной стороны является динамическим (растягиваемым) массивом, а с другой – реализует интерфейс OutputStream.

— Адаптер между массивом и OutputStream?

— Что-то вроде того. Вот как будет выглядеть наш код.

Код

public static void main(String[] args) throws Exception

{

//запоминаем настоящий PrintStream в специальную переменную

PrintStream consoleStream = System.out;

//Создаем динамический массив

ByteArrayOutputStream outputStream = new ByteArrayOutputStream();

//создаем адаптер к классу PrintStream

PrintStream stream = new PrintStream(outputStream);

//Устанавливаем его как текущий System.out

System.setOut(stream);

//Вызываем функцию, которая ничего не знает о наших манипуляциях

printSomething();

//Преобразовываем записанные в наш ByteArray данные в строку

String result = outputStream.toString();

//Возвращаем все как было

System.setOut(consoleStream);

}

public static void printSomething()

{

System.out.println("Hi");

System.out.println("My name is Amigo");

System.out.println("Bye-bye!");

}

— А что мы будем делать с полученной строкой?

— Да что угодно. Можем, например, развернуть ее задом наперед. Тогда это будет выглядеть так:

Код

public static void main(String[] args) throws Exception

{

//запоминаем настоящий PrintStream в специальную переменную

PrintStream consoleStream = System.out;

//Создаем динамический массив

ByteArrayOutputStream outputStream = new ByteArrayOutputStream();

//создаем адаптер к классу PrintStream

PrintStream stream = new PrintStream(outputStream);

//Устанавливаем его как текущий System.out

System.setOut(stream);

//Вызываем функцию, которая ничего не знает о наших манипуляциях

printSomething();

//Преобразовываем записанные в наш ByteArray данные в строку

String **result** = outputStream.toString();

//Возвращаем все как было

System.setOut(consoleStream);

//разворачиваем строку

StringBuilder stringBuilder = new StringBuilder(result);

stringBuilder.reverse();

String reverseString = stringBuilder.toString();

//выводим ее в консоль

System.out.println(reverseString);

}

public static void printSomething()

{

System.out.println("Hi");

System.out.println("My name is Amigo");

System.out.println("Bye-bye!");

}

— Как интересно. Теперь я начинаю понемногу понимать, какие большие возможности дают эти маленькие классы.  
Спасибо за интересный урок, Билаабо.