

Если вам не удалось выполнить задание в точном соответствии с инструкциями и полученный вами результат не соответствует эталонному, предлагаем проверить написанный вами код пошагово. Ниже мы описали все шаги, чтобы вам проще было найти несоответствия и (или) ошибки.

Вы должны были сначала создать класс `Printer`, эмулирующий работу принтера:

```
public class Printer {  
}
```

После этого вам было предложено создать переменную `queue`, в которой будут храниться тексты документов на распечатку, а также переменные для хранения количества документов, находящихся в данный момент в очереди на печать, а также общего количества уже распечатанных документов:

```
public class Printer {  
    private String queue = "";  
    private int pagesInQueue;  
    private int printedPages;  
}
```

Далее вы должны были создать четыре метода `append` с разными наборами параметров. Параметры предлагались такие: текст документа, имя документа и количество страниц. При этом текст документа должен был быть обязательным параметром.

Метод со всеми тремя параметрами был приведён в качестве примера:

```
public void append(String text, String name,  
    int pageCount) {  
    queue = queue + "\n" + name + " - " + text;  
    pagesInQueue = pagesInQueue + pageCount;  
}
```

По аналогии вы должны были создать метод с одним параметром (только текстом):

```
public void append(String text) {  
    queue = queue + "\nДокумент без имени - " + text;  
    pagesInQueue = pagesInQueue + 1;  
}
```

...метод с двумя параметрами — текстом и именем документа:

```
public void append(String text, String name) {
    queue = queue + "\n" + name + " - " + text;
    pagesInQueue = pagesInQueue + 1;
}
```

...и с двумя другими параметрами — текстом и количеством страниц:

```
public void append(String text, int pageCount) {
    queue = queue + "\nДокумент без имени - " + text;
    pagesInQueue = pagesInQueue + pageCount;
}
```

После этого вам было нужно создать метод, очищающий очередь печати:

```
public void clear() {
    pagesInQueue = 0;
    queue = "";
}
```

...а затем создать метод, условно распечатывающий документы из очереди и очищающий её после распечатки. Обратите внимание, что в этом методе должно происходить увеличение значения переменной `printedPages`:

```
public void print() {
    printedPages = printedPages + pagesInQueue;
    System.out.println(queue);
    clear();
}
```

Кроме того, вам нужно было создать методы получения значения переменных-счётчиков:

```
public int getPendingPagesCount() {
    return pagesInQueue;
}

public int getPrintedPagesCount() {
    return printedPages;
}
```

Затем вам предлагалось проверить работу написанного кода, скопировав следующий код в метод `main` класса `Main`:

```
Printer printer = new Printer();  
printer.append("some text 133", "Справка");  
printer.append("some text 210", 544);  
System.out.println(printer.getPendingPagesCount()); // 545  
printer.print();  
printer.append("some text 8", 530);  
printer.clear();  
printer.append("some text 23", "Договор", 312);  
printer.append("some text 11");  
printer.print();  
System.out.println(printer.getPrintedPagesCount()); // 858
```

Если вы повторите все вышеописанные шаги (сверите и исправите свой код) и запустите программу, то получите следующий результат:

```
545  
  
Справка - some text 133  
Документ без имени - some text 210  
  
Договор - some text 23  
Документ без имени - some text 11  
858
```

Если всё получилось, поздравляем! Теперь вы умеете создавать классы с переопределёнными методами и использовать их. Кроме того, в этом задании вы смогли попробовать написать значительно больше кода, чем в предыдущих заданиях.