

## Задание 1. Создайте класс с перегруженными методами

1. В этом задании вам предстоит проделать серьезную работу и написать намного больше кода, чем вы писали в предыдущих заданиях. Это необходимо для отработки навыков написания кода, которые обязательно потребуются вам в будущем.
2. В предыдущем задании вы работали в проекте Methods. Удалите из метода `main` класса `Main` всё содержимое. Продолжайте работать в этом же проекте.
3. Создайте класс `Printer`. Он будет эмулировать работу реального принтера.
4. Создайте в этом классе переменную `String queue`, в которой будет храниться список документов на печать (смотрите ниже): `String queue = "";`
5. Создайте переменные, в которых будет храниться количество страниц, находящихся в очереди на печать, и общее количество уже распечатанных страниц.
6. Создайте несколько методов `append`, принимающих документы (условно — на печать). Возможные свойства документов: текст документа, имя документа, количество страниц. Обязательным параметром является только текст документа, поэтому необходимо создать четыре метода, принимающих один, два (текст и имя либо текст и количество страниц) или три параметра.
7. При добавлении документов методами `append` они должны добавляться в строку `String queue`. Пример такого добавления:

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

Обратите внимание на символ `\n`, добавляемый в строку. Это символ переноса строки. Благодаря ему вывод в консоль содержимого переменной `queue` будет выглядеть аккуратно.

Количество страниц при этом должно добавляться к соответствующей переменной только в том случае, если оно указано в методе. Если же оно в методе не указано, то добавляется одна страница.

Если не указано имя документа, подставьте вместо него текст «Документ без имени».

8. Создайте в классе метод `clear`, очищающий очередь печати (строку `String queue`), например, так: `queue = ""`; Метод также должен обнулять количество страниц, находящихся в очереди.
9. Создайте в классе метод `print`, выводящий всю информацию о добавленных в принтер документах в консоль и очищающий очередь печати после этого (он может для этого просто вызывать уже созданный вами метод `clear`).
10. Создайте метод `getPendingPagesCount`, возвращающий общее количество страниц, которые добавлены в принтер, но ещё не распечатаны.
11. Создайте метод, возвращающий общее количество распечатанных страниц за всё время существования объекта класса `Printer`. Имя метода придумайте самостоятельно.
12. В методе `main` класса `Main` создайте объект класса `Printer` и напишите следующий код:

```
Printer printer = new Printer();
printer.append("some text 133", "Справка");
printer.append("some text 210", 544);
System.out.println(printer.getPendingPagesCount());
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());
```

13. Запустите получившийся код и убедитесь, что в консоль выводится следующий результат:

```
545
```

```
Справка - some text 133
Документ без имени - some text 210

Договор - some text 23
Документ без имени - some text 11
858
```

14. Если что-то не получилось, постарайтесь самостоятельно внести исправления в свой код и добиться необходимого результата — вывода в консоль текста, полностью идентичного показанному выше.
15. В случае возникновения трудностей, вы, как и всегда в заданиях для самостоятельного выполнения, можете воспользоваться рекомендациями под видео, под которым была указана ссылка на это задание.
16. Если всё получилось, поздравляем! Теперь вы умеете создавать классы с переопределёнными методами и использовать их. В этом задании вы проделали серьёзную работу и написали намного больше кода, чем в предыдущих заданиях. Мы специально дали вам возможность написать код повышенной сложности, чтобы в будущем вам было проще в работе.