Аргументы приложения Логирование

- **Аргументы командной строки** — это необязательные строковые аргументы, передаваемые операционной системой в программу при её запуске
- Любое, даже минимальное, приложение на Java может принимать аргументы
- Обработка и значение строковых аргументов полностью лежит на самом приложении
- Аргументы могут иметь полную и сокращенную форму записи, например: `-h` и `--help`

Передача аргументов

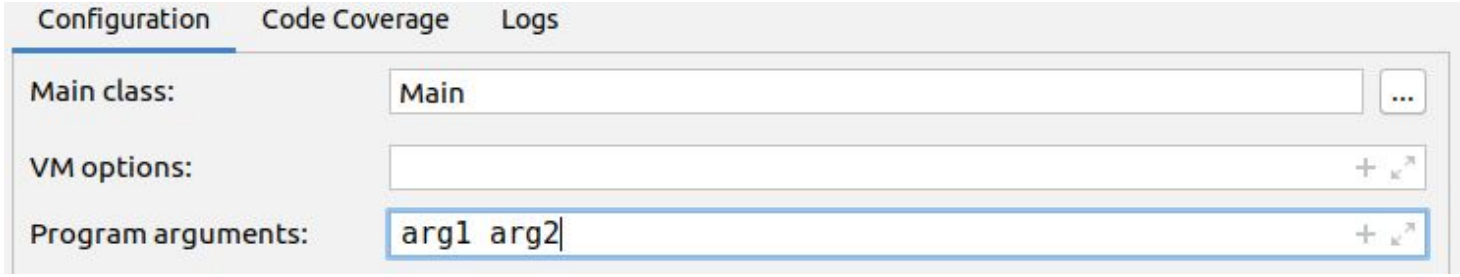
- Для java: в конце командной строки, через пробел

```
java App arg1 arg2...
```

- Для gradle: с помощью аргумента *args*

```
gradle run --args='arg1 arg2...'
```

- В IDEA: в поле *Program arguments*



```
import simple_lib.BackEnd;
public class Main {
    public static void main(String[] args) {
        if (args.length == 2)
            System.out.println("Area: " +
                BackEnd.areaCalculation(
Float.parseFloat(args[0]), Float.parseFloat(args[1])));
        else
            System.out.println("No args");
    }
}
```

$ java -cp out Main 3 2

Area: 15.7075

$ gradle run --args='3 2'

> Task :run

Area: 15.7075

BUILD SUCCESSFUL in 825ms

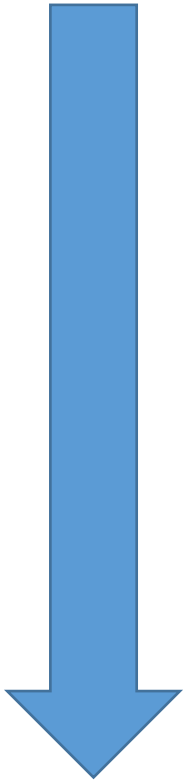
2 actionable tasks: 1 executed, 1 up-to-date

```
import simple_lib.BackEnd;
import org.apache.commons.cli.*;
public class Main {
    public static void main(String[] args) {
        Options options = new Options();
        options.addOption(Option.builder("1").hasArg().required().build());
        options.addOption(Option.builder("2").hasArg().required().build());
        CommandLineParser parser = new DefaultParser();
        HelpFormatter formatter = new HelpFormatter();
        CommandLine cmd;
        try {
            cmd = parser.parse(options, args);
        } catch (ParseException e) {
            formatter.printHelp("-1=<ARG> -2=<ARG>", options);
            return;
        }
        System.out.println("Area: " + BackEnd.areaCalculation(
                Float.parseFloat(cmd.getOptionValue("1")),
                Float.parseFloat(cmd.getOptionValue("2"))));
    }
}
```

Логирование

- Управляет сообщениями приложения
- Задает сообщениям уровень значимости
- Позволяет фильтровать сообщения по значимости
- Позволяет выбирать направление отправки сообщений

Уровни логирования



- OFF - выключено
- SEVERE - ошибки
- WARNING - предупреждения
- INFO - информация
- FINE
- FINER
- FINEST

отладка

- ALL - все

Подключение к проекту

```
public static Logger log = Logger.getLogger(App.class.getName());
Handler consoleHandler = new ConsoleHandler();
log.addHandler(consoleHandler);
consoleHandler.setLevel(Level.INFO);
log.info("Message");
```