

Практическая работа 0

«Базовая структура проекта»

Цель работы:

1. Научиться подготовить рабочее пространство перед выполнением практики
2. Научиться компилировать, собирать и выполнять проект на Java
3. Изучить Java Code Conventions

Часть 1. Подготовка среды

Для разработки программ на Java вам понадобится **JDK 8**, которую можно скачать с <https://www.oracle.com/java/technologies/javase/javase-jdk8-downloads.html> После установки убедитесь что JDK есть в переменную окружения PATH (Инструкция <https://www.javatpoint.com/how-to-set-path-in-java>)

Установите IDE для Java (Например, Eclipse, NetBeans, IntelliJ IDEA). Мы рекомендуем IntelliJ IDEA Community Edition (<https://www.jetbrains.com/ru-ru/idea/download/#section=windows>)

Установите Maven (<https://maven.apache.org/download.cgi>) Необходимо скачать, распаковать и добавить в переменную PATH/

Часть 2. Ознакомление

Данная часть предназначена для знакомства с компиляцией и сборкой программ на языке Java. Задания следует выполнять самостоятельно, результаты не предоставляются на проверку.

→ Откройте окно консоли и запустите команду компиляции `javac`. Ознакомьтесь с показанной на экране справкой.

Если вы получили сообщение о том, что такая команда не найдена, то проверьте, существует ли в переменной среды PATH путь к папке bin вашей установки JDK, если нет добавьте этот путь в переменную среды.

Настройте ваш редактор на автоматическую замену символа табуляции на 4 пробельных символа (такая настройка существует практически в любом редакторе).

→ Создайте в любой папке файл *Example1.java* и наберите в нем следующий код:

```
class Example1 {  
    public static void main(String[] args) {  
        System.out.println("Hello, World!");  
        for (String arg : args)  
            System.out.println(arg);  
    }  
}
```

→ Откомпилируйте файл используя команду `javac Example1.java`

→ Выполните программу используя команду `java Example1`, обратите внимание на то, что при выполнении указывается имя класса, а не файла (т.е. без расширения).

→ Выполните программу используя команду `java Example1 arg1 arg2 arg3`.

Часть 3. Структура проекта

→ Ознакомьтесь с требованиями к оформлению работ на <https://lms.edu-netcracker.com>

→ Создайте новую папку для проекта, содержимое этой папки будет предоставлено вами для проверки выполнения работы.

→ Скачайте и распакуйте шаблон для проекта *practice.zip*.

→ Создайте в папке для исходных кодов файл

src/main/java/com/nc/edu/ta/yourname/pr0/MainClass.java содержащий:

```
package com.nc.edu.ta.yourname.ta.pr0;

import com.nc.edu.ta.yourname.pr0.operations.Operation;

public class MainClass {
    public static void main(String[] args) {
        for (int x = 1; x < 10; x++) {
            for (int y = 1; y < 10; y++) {
                Operation operation = new Operation(x, y);
                System.out.printf("%3d", operation.getResult());
            }
            System.out.println();
        }
    }
}
```

→ Создайте в папке для исходных кодов файл

src/main/java/com/nc/edu/ta/yourname/pr0/operations/Operation.java:

```
package com.nc.edu.ta.yourname.pr0.operations;

public class Operation {

    /**
     * Create new operation using two numbers.
     *
     * @param x first operand
     * @param y second operand
     */
    public Operation(int x, int y) {
        this.x = x;
        this.y = y;
    }

    /**
     * @return result of operation
     */
}
```

```

    public int getResult() {
        return x * y;
    }

    private int x;
    private int y;
}

```

→ Запустите `run.bat` – он содержит команды сборки и запуска проекта через MAVEN. Разберитесь, что происходит.

→ **Дополнительное задание:** (необязательное) создайте в папке проекта командный файл `compile.cmd`, в котором укажите команду компиляции исходных кодов проекта в папку для откомпилированных классов. Для этого ознакомьтесь с параметрами запуска компилятора `-d`, `-classpath`, `-sourcepath`. Откомпилируйте ваши классы используя этот файл. Запустите проект в этом случае

→ **Дополнительное задание:** (необязательное) создайте в папке проекта папку `docs` и командный файл `docs.cmd`, который используя команду `javadoc` будет генерировать документацию по вашим классам в созданной папке.

Часть 4. Модульное тестирование

→ Создайте в папке исходных кодов файл

`src/test/java/com/edu/ta/yourname/pr0/operations/OperationTest.java` с текстом:

```

package com.nc.edu.ta.yourname.pr0;

import com.nc.edu.ta.yourname.pr0.operations.Operation;
import org.junit.jupiter.api.Test;

import static org.junit.jupiter.api.Assertions.assertEquals;

public class OperationTest {
    @Test
    public void testMultiply() {
        for (int a = 0; a < 100; a++)
            for (int b = 0; b < 100; b++)
                assertEquals(a + b, new Operation(a, b).getResult());
    }
}

```

→ Выполните в командной строке `mvn clean test`

→ Замените `assertEquals(a + b`, на `assertEquals(a * b`, и повторно выполните `mvn clean test`

Часть 5. IDE

Попробуйте повторить те же действия в IDE. Для этого запустите проект и создайте его на месте (поверх) вашей папки с файлами. Для запуска основного кода или теста – перейдите в соответствующий файл и нажмите зеленую стрелку возле строки объявления класса.

Часть 6. Java Code Conventions

Ознакомьтесь с [конвенцией оформления Java кода](#), обратите внимание на отступы, имена и пробелы.

Убедитесь, что ваш код соответствует этим правилам.

Часть 7. Следующие шаги

→ Создайте архив с вашими результатами. Архив должен содержать всю структуру созданных Вами каталогов и файлы *.bat, *.java, *.xml. (*.class, *.iml – не нужны)

→ Загляните в *pot.xml*. Для следующих практик замените `<practice.id>0</practice.id>` на номер практики.

→ Для следующих практик откройте любой java файл в IDE. Сделайте правый клик на строке `package com.nc.edu.ta.yourname.pr0;` В контекстном меню есть пункт `refactoring` – возможность массово переименовать пакет и перенести файл в нужную папку.