

1. Создать проект из трёх классов (основной с точкой входа и два класса в другом пакете), которые вместе должны составлять одну программу, позволяющую производить четыре основных математических действия и осуществлять форматированный вывод результатов пользователю;

Создана программа-калькулятор:

Результат работы программы:

Введите целое число: **5**

Введите арифметическое действие (+, -, *, /): **+**

Введите целое число: **6**

5 + 6 = 11

Содержание файла **Main.java**:

```
package calculate.jcore.start;

import jcore.calc.Calcluate;
import jcore.calc.Dialog;

public class Main {
    /**
     * Основной метод являющийся точкой входа
     * @param args
     */
    public static void main(String[] args) {
        Calcluate calcluate = new Calcluate();
        System.out.println(calcluate.runCalculate());
    }
}
```

Содержание файла **Calculate.java**

package calculate.jcore.calc;

```
public class Calcluate {
    protected int a;
    protected int b;
    protected int res;
    protected String act;

    Dialog dialog = new Dialog();

    /**
     * Программа калькулятора, возвращает String
     * @return
     */
    public String runCalculate() {
        a = dialog.invaiteForEnterNum();
        act = dialog.inviteForEnterAction();
        b = dialog.invaiteForEnterNum();
        switch (act) {
            case "+":
                this.res = a + b;
                break;
            case "-":
                this.res = a - b;
                break;
            case "*":
                this.res = a * b;
                break;
            case "/":
                this.res = a / b;
                break;
            default:
                this.res = -2198345;
                break;
        }
        dialog.closeScan();
        if(res == -2198345) return "До свидания";
        else return String.format("%d %s %d %s %d", a,act,b,"=",res);
    }
}
```

Содержание файла **Dialog.java**:

package calculate.jcore.calc;

import java.util.Scanner;

public class Dialog {

Scanner scanner;

/**

* Метод ввода с консоли - возвращает Integer

* @param prompt

* @return

*/

public int invaiteForEnterNum(){

System.out.print("Введите целое число: ");

int num = 0;

scanner = new Scanner(System.in);

try {

num = Integer.parseInt(scanner.nextLine());

}catch (Exception e) {

System.out.println("Введено не корректное значение.");

}

return num;

}

/**

* Метод ввода с консоли, возвращает String

* @return

*/

public String inviteForEnterAction(){

String strAct = "";

System.out.print("Введите арифметическое действие (+, -, *, /): ");

scanner = new Scanner(System.in, "cp866");

try {

strAct = scanner.nextLine();

}catch (Exception e) {

System.out.println("");

}

return strAct;

}

public void closeScan(){

scanner.close();

}

}

2. Скомпилировать проект, а также создать для этого проекта стандартную веб-страницу с документацией ко всем пакетам;

Структура каталогов для программы Калькулятор:

В пакете start – точка входа Main.java, и

в отдельном пакете calc – классы Calculate.java и Dialog.java

```
gband@gb-ubuntu:~/dzlection1$ tree
```

```
.
├── calculate
│   ├── jcore
│   │   ├── calc
│   │   │   ├── Calcluate.java
│   │   │   └── Dialog.java
│   │   ├── start
│   │   └── Main.java
└── out
```

Компиляция проекта запущена командой:

```
gband@gb-ubuntu:~/dzlection1$ javac -sourcepath ./calculate -d out
calculate/jcore/start/Main.java
```

В результате в каталоге **out** скомпилированы файлы с расширением .class:

```
gband@gb-ubuntu:~/dzlection1/out$ tree
```

```
.
├── jcore
│   ├── calc
│   │   ├── Calcluate.class
│   │   └── Dialog.class
│   ├── start
│   └── Main.class
```

Запуск проекта командой:

```
d@gb-ubuntu:~/dzlection1$ java -classpath ./out jcore/start/Main
```

```
gband@gb-ubuntu:~/dzlection1$ java -classpath ./out jcore/start/Main
Введите целое число: 10
Введите арифметическое действие (+, -, *, /): /
Введите целое число: 5
10 / 5 = 2
gband@gb-ubuntu:~/dzlection1$ |
```

Для создания документации используется команда Javadoc:

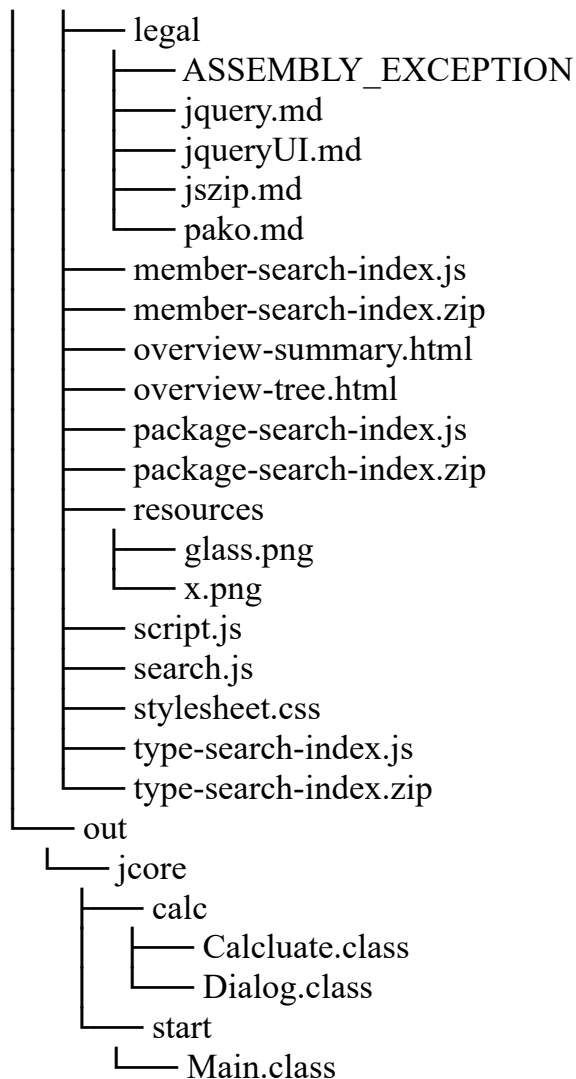
```
gband@gb-ubuntu:~/dzlection1$ javadoc -d docs -sourcepath calculate -cp out -
subpackages jcore
```

```
gband@gb-ubuntu:~/dzlection1$ javadoc -d docs -sourcepath calculate
-cp out -subpackages jcore
Loading source files for package jcore...
Constructing javadoc information
```

В результате выполнения этой команды автоматически формируется документация в каталоге **docs**:

```
gband@gb-ubuntu:~/dzlection1$ tree
```

```
.
├── calculate
│   ├── jcore
│   │   ├── calc
│   │   │   ├── Calcluate.java
│   │   │   └── Dialog.java
│   │   └── start
│   │       └── Main.java
├── docs
│   ├── allclasses.html
│   ├── allclasses-index.html
│   ├── allpackages-index.html
│   ├── constant-values.html
│   ├── deprecated-list.html
│   ├── element-list
│   ├── help-doc.html
│   ├── index-all.html
│   ├── index.html
│   ├── jcore
│   │   ├── calc
│   │   │   ├── Calcluate.html
│   │   │   ├── Dialog.html
│   │   │   ├── package-summary.html
│   │   │   └── package-tree.html
│   │   └── start
│   │       ├── Main.html
│   │       ├── package-summary.html
│   │       └── package-tree.html
│   ├── jquery
│   │   ├── external
│   │   │   └── jquery
│   │   │       └── jquery.js
│   │   ├── jquery-3.6.1.min.js
│   │   ├── jquery-ui.min.css
│   │   ├── jquery-ui.min.js
│   │   ├── jszip
│   │   │   └── dist
│   │   │       ├── jszip.js
│   │   │       └── jszip.min.js
│   │   ├── jszip-utils
│   │   │   └── dist
│   │   │       ├── jszip-utils-ie.js
│   │   │       ├── jszip-utils-ie.min.js
│   │   │       ├── jszip-utils.js
│   │   │       └── jszip-utils.min.js
│   └── jquery-ui.overrides.css
```



3. Создать Makefile с задачами сборки, очистки и создания документации на весь проект.

Создание:

gband@gb-ubuntu:~/dzlection1\$ nano Makefile

SRCDIR := calculate

OUTDIR := out

SUBPKG := jcore

all:

javac -sourcepath ./\${SRCDIR}/ -d \${OUTDIR}
\${SRCDIR}/\${SUBPKG}/start/Main.java

clean:

rm -R \${OUTDIR}/ && mkdir out

run:

cd \${OUTDIR}/ && java Main

doc:

javadoc -d docs -sourcepath \${SRCDIR} -cp out -subpackages \${SUBPKG}

В результате, можно использовать команды:

Компиляция проекта – **make**

Очистка результатов компиляции – **make clean**

Запуск проекта – **make run**

Документирование проекта – **make doc**

4. *Создать два Docker-образа. Один должен компилировать Java-проект обратно в папку на компьютере пользователя, а второй забирать скомпилированные классы и исполнять их.

Для упрощения был использован **docker compose**.

Создание первого docker-compose-file:

```
gband@gb-ubuntu:~/dzlection1$ nano docker-compose-class.yml
```

services:

app:

image: bellsoft/liberica-openjdk-alpine:11.0.16.1-1

command: javac -sourcepath ./src -d out ./src/jcore/start/Main.java

volumes:

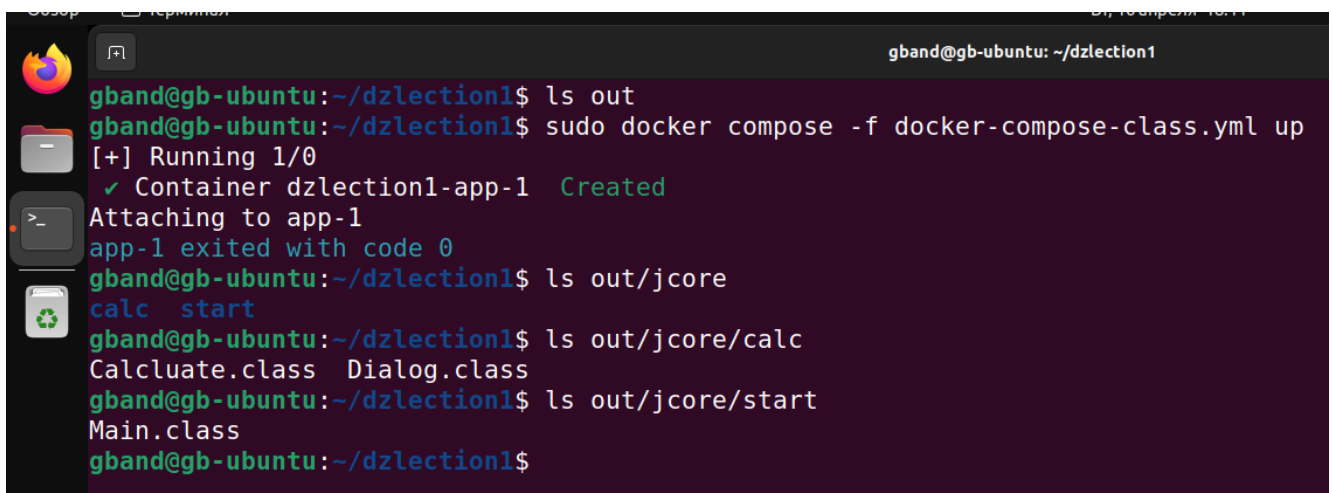
- ./out:/out

- ./calculate:/src

С помощью команды:

```
docker compose -f docker-compose-class.yml up
```

создается докер контейнер который компилирует проект в каталог **out**:



```
gband@gb-ubuntu:~/dzlection1$ ls out
gband@gb-ubuntu:~/dzlection1$ sudo docker compose -f docker-compose-class.yml up
[+] Running 1/0
 ✓ Container dzlection1-app-1 Created
Attaching to app-1
app-1 exited with code 0
gband@gb-ubuntu:~/dzlection1$ ls out/jcore
calc start
gband@gb-ubuntu:~/dzlection1$ ls out/jcore/calc
Calcuate.class Dialog.class
gband@gb-ubuntu:~/dzlection1$ ls out/jcore/start
Main.class
gband@gb-ubuntu:~/dzlection1$
```

Создание второго docker-compose-file:

```
gband@gb-ubuntu:~/dzlection1$ nano docker-compose-exec.yml
```

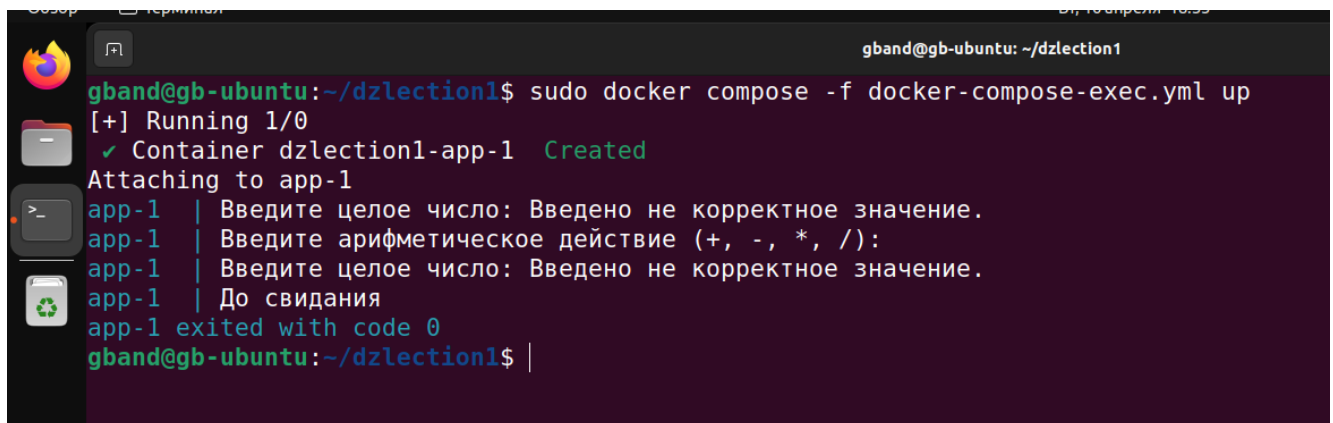
services:

```
app:
  image: bellsoft/liberica-openjdk-alpine:11.0.16.1-1
  command: java -classpath /out jcore.start.Main
  volumes:
    - ./out:/out
```

С помощью команды:

```
gband@gb-ubuntu:~/dzlection1$ sudo docker compose -f docker-compose-exec.yml
up
```

Запускается проект из каталога **out**:



```
gband@gb-ubuntu: ~/dzlection1
gband@gb-ubuntu:~/dzlection1$ sudo docker compose -f docker-compose-exec.yml up
[+] Running 1/0
 ✓ Container dzlection1-app-1 Created
Attaching to app-1
app-1 | Введите целое число: Введено не корректное значение.
app-1 | Введите арифметическое действие (+, -, *, /):
app-1 | Введите целое число: Введено не корректное значение.
app-1 | До свидания
app-1 exited with code 0
gband@gb-ubuntu:~/dzlection1$
```

5* Создать docker-контейнер для формирования полной документации по проекту.

Создание docker-compose-file:

```
gband@gb-ubuntu:~/dzlection1$ nano docker-compose-docs.yml
```

```
services:
  app:
    image: bellsoft/liberica-openjdk-alpine:11.0.16.1-1
    command: javadoc -d docs -sourcepath /src -cp /out -subpackage jcore
    volumes:
      - ./out:/out
      - ./calculate:/src
```

С помощью команды:

```
gband@gb-ubuntu:~/dzlection1$ sudo docker compose -f docker-compose-docs.yml
up
```

Запускается документирование проекта :


```
gband@gb-ubuntu: ~/dzlection1
gband@gb-ubuntu:~/dzlection1$ ls docs
gband@gb-ubuntu:~/dzlection1$ sudo docker compose -f docker-compose-docs.yml up
[sudo] пароль для gband:
[+] Running 0/0
  Container dzlection1-app-1 Created
Attaching to app-1
app-1 | Loading source files for package jcore...
app-1 | Constructing Javadoc information...
app-1 | Standard Doclet version 11.0.16.1
app-1 | Building tree for all the packages and classes...
app-1 | Generating docs/jcore/calc/Calcluate.html...
app-1 | /src/jcore/calc/Calcluate.java:13: warning: no description for @return
app-1 | * @return
app-1 | ^
app-1 | Generating docs/jcore/calc/Dialog.html...
app-1 | /src/jcore/calc/Dialog.java:10: error: @param name not found
app-1 | * @param prompt
app-1 | ^
app-1 | /src/jcore/calc/Dialog.java:10: warning: no description for @param
app-1 | * @param prompt
app-1 | ^
app-1 | /src/jcore/calc/Dialog.java:11: warning: no description for @return
app-1 | * @return
app-1 | ^
app-1 | /src/jcore/calc/Dialog.java:26: warning: no description for @return
app-1 | * @return
app-1 | ^
app-1 | Generating docs/jcore/start/Main.html...
app-1 | /src/jcore/start/Main.java:9: warning: no description for @param
app-1 | * @param args
app-1 | ^
app-1 | Generating docs/jcore/calc/package-summary.html...
```

```
gband@gb-ubuntu:~/dzlection1$ ls docs
allclasses.html      help-doc.html      member-search-index.zip  script.js
allclasses-index.html  index-all.html    overview-summary.html    search.js
allpackages-index.html index.html          overview-tree.html       stylesheet.css
constant-values.html  jcore              package-search-index.js  type-search-index.js
deprecated-list.html  jquery             package-search-index.zip  type-search-index.zip
element-list          member-search-index.js resources
gband@gb-ubuntu:~/dzlection1$
```