

[Каталог курсов](#)[Сотрудничество](#)[О нас](#)[Поиск в каталоге](#)[Мой профиль](#)

[Курс](#) > [Итоговое тестирование](#) > [Итоговое тестирование](#) > Итоговое тестирование

Итоговое тестирование

Этот элемент курса оценивается как 'Итоговое тестирование'

вес: 1.0

Blank Advanced Problem

14/14 points (graded)

1. Что произойдёт при сборке проекта с приведённым ниже Makefile в случае указания цели “run”?

```
JFLAGS = -g
JC = javac
JVM= java
FILE=

.SUFFIXES: .java .class

.java.class:
    $(JC) $(JFLAGS) $*.java

CLASSES = \
    Experiment.java \
    Block.java \
    Spring.java \
    PhysicsElement.java \
    Simulator.java
```

```
MAIN = Experiment
```

```
default: classes
```

```
classes: $(CLASSES:.java=.class)
```

```
run: $(MAIN).class  
      $(JVM) $(MAIN)
```

```
clean:  
      $(RM) *.class
```

(один вариант ответа)

☐ Будет удалён файл Experiment.java.

☐ Будет удалён файл Block.class.

☐ Будет удалён файл Autumn.class.

☐ Будет запущен исполняемый класс Simulator.

☒ Будет запущен исполняемый класс Experiment.

☐ Будет скомпилирован код из файла Spring.java.

☐ Будет скомпилирован код из файла Autumn.java.

☐ Будет вызван компилятор javac с флагом -g.



2. Системный администратор распаковал архив программы, и ему известно, что разработчик использовал autotools для автоматической конфигурации ПО. Какая обычная последовательность действий для сборки и установки программы? (один вариант ответа)

☒ ./configure; make; sudo make install

☐ ./configure; make;

☐ automake; ./configure; make; sudo make install

☐ autoconf; automake; ./configure; sudo make install

☐ ./configure; automake; sudo make install

☐ ./configure; sudo automake;



3. Что произойдёт при сборке проекта с приведённым ниже build.xml в случае указания цели "docs"?

```
<?xml version="1.0"?>
<project name="Ant-Test" default="main" basedir=".">
  <property name="src.dir" location="src" />
  <property name="build.dir" location="bin" />
  <property name="dist.dir" location="dist" />
  <property name="docs.dir" location="docs" />

  <target name="clean">
    <delete dir="${build.dir}" />
    <delete dir="${docs.dir}" />
    <delete dir="${dist.dir}" />
  </target>

  <target name="mkdir">
    <mkdir dir="${build.dir}" />
    <mkdir dir="${docs.dir}" />
    <mkdir dir="${dist.dir}" />
  </target>

  <target name="compile" depends="clean, mkdir">
    <javac srcdir="${src.dir}" destdir="${build.dir}">
    </javac>
  </target>

  <target name="docs" depends="compile">
    <javadoc packagenames="src" sourcepath="${src.dir}"
      destdir="${docs.dir}">
      <fileset dir="${src.dir}">
        <include name="**" />
      </fileset>
    </javadoc>
  </target>

  <target name="jar" depends="compile">
```

```
<jar destfile="${dist.dir}\dispell-all-students.jar"
      basedir="${build.dir}">
  <manifest>
    <attribute name="Main-Class" value="test.Main" />
  </manifest>
</jar>
</target>

<target name="main" depends="compile, jar, docs">
  <description>Main target</description>
</target>

</project>
```

(несколько вариантов ответа)

☐ Будет удалён каталог \$PROJECT_HOME/bin.

☐ Будет создан каталог \$PROJECT_HOME/bin.

☐ Будет удалён каталог \$PROJECT_HOME/build.

☐ Будет создан каталог \$PROJECT_HOME/build.

☐ В каталоге \$PROJECT_HOME/dist будет создан исполняемый архив dispell-all-students.jar.

☒ Будет вызвана утилита javadoc, которая автоматически сформирует документацию на основании javadoc-комментариев в исходном коде.

☒ Документация по проекту будет перемещена в каталог \$PROJECT_HOME/docs.

☐ Будет скомпилирован код, хранящийся в каталоге \$PROJECT_HOME/src.



4. Сколько целей будет собрано, если запустить сборку проекта с приведённым ниже build.xml посредством команды “ant clean”?


```
<?xml version="1.0"?>
<project name="Ant-Test" default="main" basedir=".">
  <property name="src.dir" location="src" />
  <property name="build.dir" location="bin" />
  <property name="test.dir" location="src" />
  <property name="test.report.dir" location="testreport" />

  <path id="junit.class.path">
    <pathelement location="lib/junit.jar" />
    <pathelement location="${build.dir}" />
  </path>

  <target name="clean">
    <delete dir="${build.dir}" />
    <delete dir="${test.report.dir}" />
  </target>

  <target name="mkdir">
    <mkdir dir="${build.dir}" />
    <mkdir dir="${test.report.dir}" />
  </target>

  <target name="compile" depends="clean, mkdir">
    <javac srcdir="${src.dir}" destdir="${build.dir}">
      <classpath refid="junit.class.path" />
    </javac>
  </target>

  <target name="junit" depends="compile">
    <junit printsummary="on" fork="true" haltonfailure="yes">
      <classpath refid="junit.class.path" />
      <formatter type="xml" />
      <batchtest todir="${test.report.dir}">
        <fileset dir="${src.dir}">
```

```
        <include name="**/*Test*.java" />
    </fileset>
</batchtest>
</junit>
</target>

<target name="main" depends="compile, junit">
    <description>Main target</description>
</target>

</project>
```

(один вариант ответа)

☐ Ни одной.

☒ Одна. ✓

☐ Две.

☐ Три.

☐ Более трёх.

5. В каком порядке будут вызваны плагины в случае сборки проекта с pom.xml, приведённым ниже, командой "mvn clean package"?

```
(...)  
<plugin>  
  <groupId>org.apache.maven.plugins</groupId>  
  <artifactId>maven-checkstyle-plugin</artifactId>  
  <version>2.6</version>  
  <executions>  
    <execution>  
      <phase>test</phase>  
      <goals>  
        <goal>check</goal>  
      </goals>  
    </execution>  
  </executions>  
</plugin>  
<plugin>  
  <groupId>ru.ifmo.dispell</groupId>  
  <artifactId>student-dispeller-plugin</artifactId>  
  <version>2019.1</version>  
  <executions>  
    <execution>  
      <phase>package</phase>  
      <goals>  
        <goal>dispell-all</goal>  
      </goals>  
    </execution>  
  </executions>  
</plugin>  
<plugin>  
  <groupId>com.github.warmuuh</groupId>  
  <artifactId>libsass-maven-plugin</artifactId>  
  <version>0.2.9-libsass_3.4.7</version>  
  <executions>  
    <execution>  
      <phase>generate-sources</phase>  
      <goals>  
        <goal>compile</goal>  
      </goals>  
    </execution>  
  </executions>  
</plugin>
```

```
<configuration>
  <inputPath>${project.basedir}/static/sass</inputPath>
  <outputPath>${project.basedir}/static/css</outputPath>
  <outputStyle>expanded</outputStyle>
</configuration>
</plugin>
(...)
```

(один вариант ответа)

- ☐ maven-checkstyle-plugin, libsass-maven-plugin, student-dispeller-plugin.
- ☐ maven-checkstyle-plugin, student-dispeller-plugin, libsass-maven-plugin.
- ☐ student-dispeller-plugin, libsass-maven-plugin, maven-checkstyle-plugin.
- ☐ student-dispeller-plugin, maven-checkstyle-plugin, libsass-maven-plugin.
- ☐ libsass-maven-plugin, student-dispeller-plugin, maven-checkstyle-plugin.
- ☒ libsass-maven-plugin, maven-checkstyle-plugin, student-dispeller-plugin. ✓

6. Что будет выведено в случае сборки проекта с build.gradle, приведённым ниже, командой "gradle -q the_evil"?

```
defaultTasks 'the_good', 'the_bad'
```

```
task the_good {  
    doLast {  
        print "I'm good! "  
    }  
}
```

```
task the_bad {  
    doLast {  
        print "I'm bad! "  
    }  
}
```

```
task the_evil {  
    doLast {  
        print "I'm evil! "  
    }  
}
```

(один вариант ответа)

☐ I'm good!

☐ I'm bad!

☒ I'm evil! ✓

☐ I'm good! I'm bad!

☐ I'm bad! I'm good!

☐ I'm good! I'm bad! I'm evil!

7. К какой категории проблемных ситуаций жизненного цикла ПО относится приведённый ниже пример?

```
install.sh
... @@ -37,7 +37,7 @@
37 37 # You should have received a copy of the GNU General Public License
38 38 # along with bumblebee. If not, see <http://www.gnu.org/licenses/>.
39 39 #
40 -BUMBLEBEEVERSION=1.4.31
40 +BUMBLEBEEVERSION=1.4.32
41 41
42 42
43 43 ROOT_UID=0
... @@ -348,7 +348,7 @@ case "$DISTRO" in
348 348 ln -s /usr/lib/mesa/ld.so.conf /etc/alternatives/gl_conf
349 349 rm -rf /etc/alternatives/xorg_extra_modules
350 350 rm -rf /etc/alternatives/xorg_extra_modules-bumblebee
351 - rm -rf /usr/lib/nvidia-current/xorg/xorg
```

(один вариант ответа)

☐ Mistake.

☒ Fault. ✓

☐ Failure.

☐ Error.

8. Какие типы тестирования подходят наилучшим образом для решения задачи, возникающей в описанной ниже ситуации?
“Информационная система банка “Аптекарский” написана на платформе Java EE, состоит из большого числа взаимодействующих компонентов (в первую очередь, EJB), каждый из которых реализует достаточно сложную бизнес-логику, от корректности работы которой зависит эффективность работы банка. Интерфейсы компонентов специфицированы и меняются нечасто”. (несколько вариантов ответа)

☒ Модульное тестирование.

☒ Интеграционное тестирование.

☐ Статическое тестирование.

☐ Приёмочное тестирование.

☐ Стресс-тестирование.



9. Программисты Банка “Ваше Богатство” определяет тестовое покрытие для функции вычисления размера процентов на остаток по счету клиента. При наличии на счете до 100 000 руб. ежемесячно начисляется сумма эквивалентная 5.5% годовых; если на счете больше 100 000 руб. то 2%. Сколько эквивалентных участков должно содержать тестовое покрытие, если в функцию могут поступать и некорректные (например отрицательные) данные. (один вариант ответа)

☐ ни одного

☒ три ✓

☐ четыре

☐ пять

☐ шесть

☐ семь

10. Что будет выведено в результате запуска тестового сценария на JUnit из примера ниже?


```
import org.junit.*;

public class AnnotationTest1 {

    @BeforeClass
    public static void m1() {
        System.out.print("m1 ");
    }

    @AfterClass
    public static void m2() {
        System.out.print("m2 ");
    }

    @Before
    public void m3() {
        System.out.print("m3 ");
    }

    @After
    public void m4() {
        System.out.print("m4 ");
    }

    @Test
    public void m5() {
        System.out.print("m5 ");
    }

    @Test
    public void m6() {
        System.out.print("m6 ");
    }
}
```

```
}  
  
}
```

(один вариант ответа)

☐ "m1 m2 m3 m4 m5 m6 "

☐ "m1 m3 m5 m6 m4 m2 "

☐ "m3 m1 m5 m6 m2 m4 "

☒ "m1 m3 m5 m4 m3 m6 m4 m2 " ✓

☐ "m3 m1 m5 m2 m1 m6 m2 m4 "

☐ "m1 m3 m5 m4 m2 m1 m3 m6 m4 m2 "

☐ "m3 m1 m5 m2 m4 m3 m1 m6 m2 m4 "

11. Начинающий программист разбирал модульные тесты нерадивого разработчика. Ему ужасно не нравился код нерадивого разработчика. Дополнительная сложность была в том, что функция, которую тестировал программист, была скрыта в "черном ящике". Помогите начинающему программисту определить функцию, скрытую в модуле, чтобы он мог составить удовлетворительное тестовое покрытие, если известно, что при тестировании некорректно выполняются тест c4.

```
@Test public void c1() {  
    assertEquals(0d, f(-2d), 0.0001d);  
}  
  
@Test public void c2() {  
    assertEquals(0d, f(2d), 0.0001d);  
}  
  
@Test public void c3() {  
    assertEquals(-4d, f(0d), 0.0001d);  
}  
  
@Test public void c4() {  
    assertEquals(-16d, f(0d), 0.0001d);  
}  
  
@Test public void c5() {  
    assertEquals(0d, f(6d), 0.0001d);  
}
```

(один вариант ответа)

☐ $f(x) = x^4 - 16$

☐ $f(x) = 2x - 4$

☐ $f(x) = -2x - 4$

☒ $f(x) = -4\cos(\pi x/4)$ ✓

☐ $f(x) = x^2 - 4$

количество минут в невисокосный год данная система может простаивать? (один вариант ответа)

☐ 0.

☐ 5.

☒ 52. ✓

☐ 105.

☐ 420.

13. Программа вычисляет значение по алгоритму, обрабатывая данные на уровне процессора (1 команда = 1 нс), уровне кэш-памяти второго уровня (20 нс), оперативной памяти (100 нс). Для вычисления значения алгоритма используются 1000 команд на уровне процессора, 100 обращений к памяти, из которых 90% кешируются в кэш-памяти второго уровня. На сколько процентов увеличится скорость работы алгоритма, если программист смог уменьшить количество обращений к оперативной памяти до тридцати раз? (один вариант ответа)

☐ Увеличится на 166%

☒ Увеличится на 207% ✓

☐ Увеличится на 353%

☐ Увеличится на 378%

☐ Увеличится на 402%

☐ Не изменится. Так оптимизировать алгоритм нет необходимости

14. Какие показатели могут говорить о проблемах с производительностью на приведённом ниже скриншоте?

```
top - 11:40:49 up 26 days,  1:30,  1 user,  load average: 0,87, 0,79, 0,73
Tasks: 351 total,   1 running, 285 sleeping,   0 stopped,   9 zombie
%Cpu(s):  1,2 us,  0,7 sy,  0,0 ni, 98,0 id,  0,1 wa,  0,0 hi,  0,0 si,  0,0 st
КиБ Mem : 12188692 total,  131072 free,  8961588 used,  3096032 buff/cache
КиБ Swap: 3999740 total,   66032 free,  3933708 used.  212234 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1837	user	20	0	4423976	449200	84120	S	2,6	3,7	657:57.30	gnome-shell
1658	user	20	0	8803704	2230324	185440	S	1,3	1,9	474:39.65	java
4624	user	20	0	6770984	1241776	114996	S	1,0	0,3	15:24.29	java
9834	user	20	0	51500	4412	3528	R	0,7	0,0	0:00.14	top
8	root	20	0	0	0	0	I	0,3	0,0	37:57.74	rcu_sched
1247	gdm	20	0	3592052	55840	22712	S	0,3	0,5	17:16.01	gnome-shell
1319	root	-51	0	0	0	0	S	0,3	0,0	22:07.02	irq/49-nvi+
1452	gdm	20	0	889868	15288	3728	S	0,3	0,1	15:34.10	gsd-color
1587	root	20	0	1104540	1000	0	S	0,3	0,0	22:20.80	teamviewerd
8726	root	20	0	0	0	0	I	0,3	0,0	0:00.56	kworker/2:1
9537	user	20	0	2787916	2261384	113872	S	0,3	2,1	0:07.45	java
9718	user	20	0	3020820	3373248	93828	S	0,3	3,1	0:29.72	java
1	root	20	0	226180	6292	3728	S	0,0	0,1	5:22.40	systemd

(один вариант ответа)

☐ Малая доля времени, которое процессор проводит на уровне ядра ОС.

☐ Малая доля времени, которое процессор проводит на уровне пользовательского приложения.

☐ Короткая очередь процессов.

☐ Низкая загрузка процессора.

☒ Недостаточный объём свободной памяти. ✓

☐ На приведённом скриншоте нет показателей, которые могли бы свидетельствовать о проблемах с производительностью.

Отправить



[Каталог курсов](#)

[Направления подготовки](#)

[О нас](#)

[Вопросы и ответы](#)

[Пользовательское соглашение](#)

[Контакты](#)

[Помощь](#)

© 2020 Открытое образование

