МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики»

ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ

**ЛАБОРАТОРНАЯ РАБОТА №3**

по дисциплине

«Основы программной инженерии»

Вариант № 125103

***Выполнили:***

Студенты группы P3216

Векшин А. И.

Козодой А.С.

***Преподаватель:***

Письмак Алексей Евгеньевич

# 

[**Задание 3**](#_g7yvr5ee51ia)

[**build.xml 4**](#_49usfucba350)

[**build.properties 6**](#_tkdqlv5kdmvq)

[**Результаты работы 7**](#_nqi9yukr9qdx)

[**Вывод 8**](#_cyp0yd1eoiuk)

# 

# **Задание**

Написать сценарий для утилиты [Apache Ant](http://ant.apache.org), реализующий компиляцию, тестирование и упаковку в jar-архив кода проекта из [лабораторной работы №3](https://se.ifmo.ru/courses/web#lab3) по дисциплине "Веб-программирование".

Каждый этап должен быть выделен в отдельный блок сценария; все переменные и константы, используемые в сценарии, должны быть вынесены в отдельный файл параметров; MANIFEST.MF должен содержать информацию о версии и о запускаемом классе.

**Cценарий должен реализовывать следующие цели (targets):**

1. **compile** -- компиляция исходных кодов проекта.
2. **build** -- компиляция исходных кодов проекта и их упаковка в исполняемый jar-архив. Компиляцию исходных кодов реализовать посредством вызова цели **compile**.
3. **clean** -- удаление скомпилированных классов проекта и всех временных файлов (если они есть).
4. **test** -- запуск junit-тестов проекта. Перед запуском тестов необходимо осуществить сборку проекта (цель **build**).
5. **xml** - валидация всех xml-файлов в проекте.
6. **alt** - создаёт альтернативную версию программы с измененными именами переменных и классов (используя задание replace/replaceregexp в файлах параметров) и упаковывает её в jar-архив. Для создания jar-архива использует **Git**

# 

# **build.xml**

<project name="JSFProject" default="build" basedir=".">

*<!-- Загрузка файлов конфигураций -->*

<property file="build.properties"/>

*<!-- Создание целей -->*

<target name="compile">

<mkdir dir="${build.dir}"/>

<javac srcdir="${src.dir}" destdir="${build.dir}" includeantruntime="false" classpathref="project.classpath"/>

</target>

<target name="build" depends="compile">

<sound>

<fail source="eche\_raz.wav"/>

<success source="spasibo-kloun.wav"/>

</sound>

<jar destfile="${dist.dir}/${jar.name}" basedir="${build.dir}">

<fileset dir="${web.dir}">

<include name="\*\*/\*.\*"/>

</fileset>

<manifest>

<attribute name="Main-Class" value="${main.class}"/>

<attribute name="Implementation-Version" value="${version}"/>

</manifest>

</jar>

</target>

<target name="compile-tests" depends="build">

<javac srcdir="${test.dir}" destdir="${build.dir}" classpathref="project.classpath" includeantruntime="false"/>

</target>

<target name="clean">

<delete dir="${build.dir}"/>

<delete file="${dist.dir}/${jar.name}"/>

</target>

<target name="test" depends="compile-tests">

<sound>

<fail source="eche\_raz.wav"/>

<success source="spasibo-kloun.wav"/>

</sound>

<junit fork="yes" haltonfailure="yes">

<classpath refid="project.classpath"/>

<classpath location="${dist.dir}/${jar.name}"/>

<formatter type="plain"/>

<batchtest todir="${test.report.dir}">

<fileset dir="${test.dir}">

<include name="\*\*/\*Test.java"/>

</fileset>

</batchtest>

</junit>

</target>

<target name="xml">

<xmlvalidate file="${src.dir}/main/webapp/WEB-INF/web.xml" />

</target>

<target name="alt" depends="build">

<replace file="${src.dir}/main/java/PropertyClass.java" token="ненавижу" value="люблю"/>

<jar destfile="${dist.dir}/alterJar" basedir="${build.dir}">

<manifest>

<attribute name="Main-Class" value="${main.class}"/>

<attribute name="Implementation-Version" value="1.4.8.8"/>

</manifest>

</jar>

</target>

<path id="project.classpath">

<fileset dir="${lib.dir}">

<include name="jsf-api-2.1.jar"/>

<include name="jsf-impl-1.2-20.jar"/>

<include name="junit-4.13.2.jar"/>

<include name="javax.el-api-3.0.0.jar"/>

<include name="javax.servlet-api-4.0.1.jar"/>

<include name="primefaces-13.0.8.jar"/>

<include name="javax.persistence-api-2.2.jar"/>

<include name="postgresql-42.7.3.jar"/>

<include name="javax.faces-api-2.3.jar"/>

<include name="hibernate-core-6.5.0.CR2.jar"/>

<include name="hibernate-validator-8.0.1.Final.jar"/>

<include name="hamcrest-all-1.3.jar"/>

</fileset>

</path>

</project>

# 

# **build.properties**

*# Directories*

src.dir=web3/src/

build.dir=build

dist.dir=dist

lib.dir=lib

test.dir=test

test.report.dir=test-report

web.dir=web3/src/main/webapp

*# Files*

jar.name=JSFProject.jar

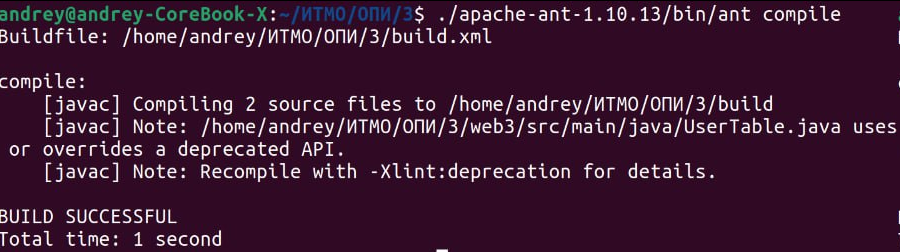
*# Classpath*

project.classpath=hibernate-core-6.5.0.CR2.jar;jsf-api-2.1.jar;hibernate-validator-8.0.1.Final.jar;jsf-impl-1.2-20.jar;javax.el-api-3.0.0.jar;junit-4.13.2.jar;javax.faces-api-2.3.jar;postgresql-42.7.3.jar;javax.persistence-api-2.2.jar;primefaces-13.0.8.jar;javax.servlet-api-4.0.1.jar

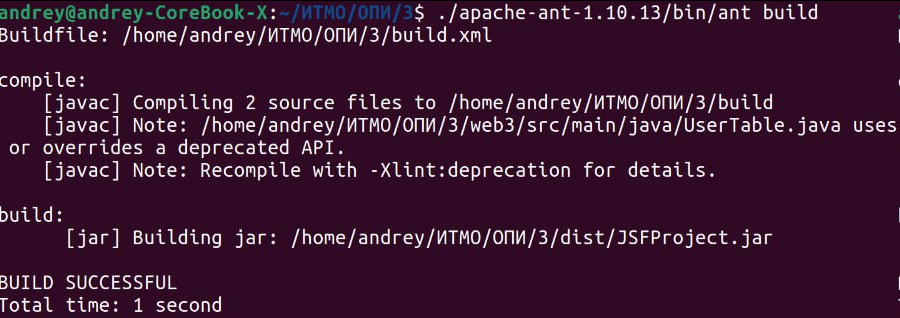
# 

# **Результаты работы**

**compile**

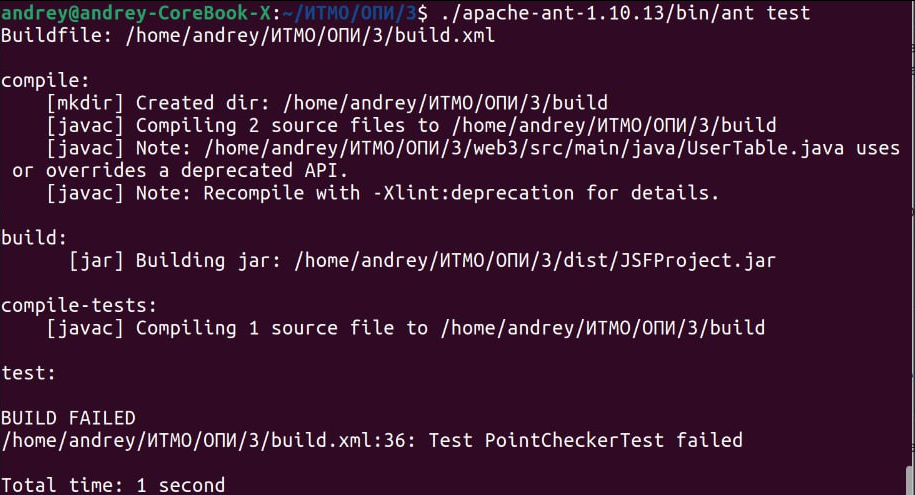


**build**

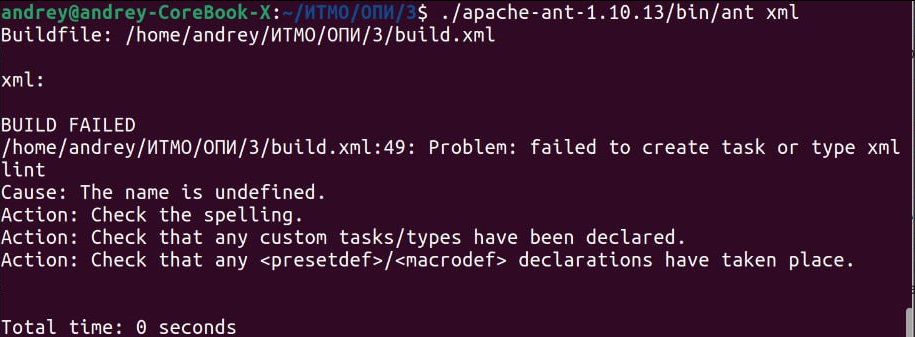


**clean**

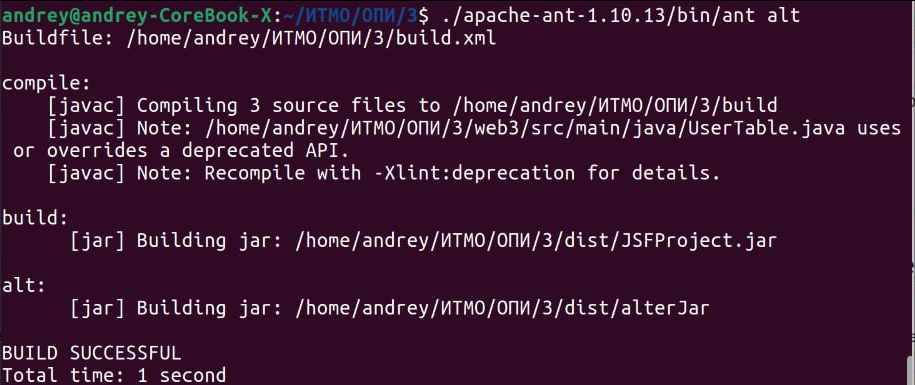


**test**

**xml**



**alt**



# **Вывод**

В ходе выполнения лабораторной работы мы ознакомились с системой автоматической сборки Apache Ant

.∧\_∧

(・ω・)つ━☆・\*。

⊂ ノ ・ ゚+.

しーJ °。+ \* ́ ̈)

.· ́ ̧.·\* ́ ̈)

( ̧.· ́ ( ̧.·'\* ☆ JSF пора на свалку истории