

# МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ»**

# МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ»**

# МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ»**

# МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ»**

**Факультет управления и информатики в технологических системах**

**Кафедра**[**корпоративных информационных систем и программирования**](https://vsuet.ru/obuchenie/faculties/uits/k_kisip)

**Направление подготовки 09.03.02**

***(шифр)***

**Информационные системы и технологии**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*(наименование направления подготовки)*

**Отчет**

**по практическому заданию № 11**

Выполнил студент гр. У-213

Боровкова Анастасия Дмитриевна

(ф.и.о)

**\_\_\_\_\_\_\_\_\_**\_\_\_\_\_\_\_\_\_\_\_\_\_\_

***(****подпись****)***

Проверил(а):

Демичев С.Е.

**\_\_\_\_\_\_\_\_\_**\_\_\_\_\_\_\_\_\_\_\_\_\_\_

***(****подпись****)***

Воронеж – 2023

public class TreePrototypesUnitTest {  
  
 @Test  
 public void givenAPlasticTreePrototypeWhenClonedThenCreateA\_Clone() {  
 *// ...*  
  
 PlasticTree plasticTree = new PlasticTree(mass, height);  
 plasticTree.setPosition(position);  
 PlasticTree anotherPlasticTree = (PlasticTree) plasticTree.copy();  
 anotherPlasticTree.setPosition(otherPosition);  
  
 assertEquals(position, plasticTree.getPosition());  
 assertEquals(otherPosition, anotherPlasticTree.getPosition());  
 }  
}

@Test  
public void givenA\_ListOfTreesWhenClonedThenCreateListOfClones() {  
  
 *// create instances of PlasticTree and PineTree*  
  
 List<Tree> trees = Arrays.asList(plasticTree, pineTree);  
 List<Tree> treeClones = trees.stream().map(Tree::copy).collect(toList());  
  
 *// ...*  
  
 assertEquals(height, plasticTreeClone.getHeight());  
 assertEquals(position, plasticTreeClone.getPosition());  
}