

# 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 № 12**

Выполнил студент гр. У-213

Боровкова Анастасия Дмитриевна

(ф.и.о)

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

***(****подпись****)***

Проверил(а):

Демичев С.Е.

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

***(****подпись****)***

Воронеж – 2023

// Единообразный интерфейс доступа к листовым и контейнерным объектам  
public interface SubExpression {  
    
  public Number value();  
    
  public void add(SubExpression expr);  
  public void sub(SubExpression expr);  
  public SubExpression getSubExpression(int index);  
}  
  
// Лист - целое число  
public class IntegerValue implements SubExpression {  
    
  private Integer value;  
    
  public IntegerValue(Integer value) {  
    this.value = value;  
  }  
  
  @Override  
  public void add(SubExpression expr) {  
    throw new UnsupportedOperationException();      
  }  
  
  @Override  
  public SubExpression getSubExpression(int index) {  
    throw new UnsupportedOperationException();  
  }  
  
  @Override  
  public void sub(SubExpression expr) {  
    throw new UnsupportedOperationException();      
  }  
  
  @Override  
  public Number value() {  
    return value;  
  }  
}  
  
import java.util.ArrayList;  
import java.util.List;  
  
// Выражение - контейнер  
public class Expression implements SubExpression {  
    
  private List<SubExpression> exprs;  
    
  public Expression(SubExpression ... exprs) {  
    this.exprs = new ArrayList<SubExpression>();  
    for (SubExpression expr: exprs) {  
      this.exprs.add(expr);  
    }  
  }  
    
  @Override  
  public void add(SubExpression expr) {  
    exprs.add(expr);      
  }  
    
  @Override  
  public void sub(SubExpression expr) {  
    if (expr instanceof IntegerValue) {  
      exprs.add(new IntegerValue(-1\*expr.value().intValue()));  
    } else {  
      exprs.add(new FloatValue(-1\*expr.value().floatValue()));  
    }  
      
  }  
  
  @Override  
  public SubExpression getSubExpression(int index) {  
    return exprs.get(index);  
  }  
  
  @Override  
  public Number value() {  
    Number result = new Float(0);  
      
    for (SubExpression expr: exprs) {  
      result = result.floatValue() + expr.value().floatValue();  
    }  
      
    return result;  
  }  
}  
  
// Использование  
public class Main {  
  
  public static void main(String[] args) {  
    // Вычислим выражение - 20 - (5-2) - (11+6)  
    // Приведем к следующему виду 20 - a - b  
    SubExpression expr = new Expression();  
  
    SubExpression a = new Expression(new IntegerValue(5), new IntegerValue(-2));  
    SubExpression b = new Expression(new IntegerValue(11), new IntegerValue(6));  
      
    expr.add(new IntegerValue(20));  
    expr.sub(a);  
    expr.sub(b);  
      
    System.out.println(expr.value());  
  }  
}