Молдавский Государственный Университет

Факультет Математики и Информатики

Департамент Информатики

Лабораторная работа №5

по курсу Java

на тему

“Generics. Обобщенные классы и методы”

Выполнил: Bogdanov Iurii,

группа I2302

Проверил: Gheorghe Latul

Кишинев, 2024

**Цель работы:**

Целью данной лабораторной работы является изучение обобщённых классов и методов в Java. Основное внимание уделяется концепциям Generics (обобщений), которые позволяют создавать классы и методы, способные работать с любым типом данных, обеспечивая при этом строгую типизацию и безопасность.

Результатом работы должно стать понимание, как обобщённые классы и методы упрощают программирование, повышают читаемость и уменьшают дублирование кода.

**Описание задачи:**

· Реализовать обобщённый класс Box<T>, который может хранить объект любого типа.

· Добавить методы для установки и получения значения (setValue, getValue).

· Реализовать обобщённый метод compareTo(T other) в классе Box, который будет сравнивать значения объектов. Тип T должен реализовывать интерфейс Comparable<T>.

· Написать демонстрационную программу, которая протестирует работу класса Box с типами Integer и String.

**Реализация:**

Для выполнения задачи был разработан обобщённый класс Box<T>, который может хранить значения любого типа, реализующего интерфейс Comparable<T>. Этот класс имеет следующие возможности:

1. Конструктор для создания объекта с заданным значением.
2. Методы setValue и getValue для установки и получения значения.
3. Метод compareTo, который сравнивает текущее значение с другим объектом того же типа T. Метод использует интерфейс Comparable для выполнения сравнения.

Класс Box<T> имеет следующий код:

public class Box<T extends Comparable<T>> {  
 private T value;  
  
 public Box() {}  
  
 public Box(T value) {  
 this.value = value;  
 }  
  
 public void setValue(T value) {  
 this.value = value;  
 }  
  
 public T getValue() {  
 return value;  
 }  
  
 public int compareTo(Box<T> other) {  
 if (value != null && other.getValue() != null) {  
 return value.compareTo(other.getValue());  
 }  
 throw new IllegalArgumentException("Values must not be null");  
 }

Ключевая особенность данного класса — использование ограничений T extends Comparable<T>. Это позволяет гарантировать, что тип T поддерживает сравнение, необходимое для работы метода compareTo.

Главный метод программы, находящийся в классе Main, демонстрирует использование обобщённого класса Box на примерах с объектами типов Integer и String. Для каждого типа создаются два объекта, значения которых сравниваются с использованием метода compareTo, а результаты выводятся на экран. Также программа использует метод toString для отображения содержимого объектов.

Код класса Main:

public class Main {  
 public static void main(String[] args) {  
  
 Box<Integer> intBox1 = new Box<>(10);  
 Box<Integer> intBox2 = new Box<>(20);  
  
 System.*out*.println("intBox1: " + intBox1);  
 System.*out*.println("intBox2: " + intBox2);  
  
 int comparisonResult = intBox1.compareTo(intBox2);  
 System.*out*.println("Сравнение intBox1 и intBox2: " +  
 (comparisonResult < 0 ? "intBox1 меньше" : comparisonResult > 0 ? "intBox1 больше" : "равны"));  
  
 Box<String> strBox1 = new Box<>("Apple");  
 Box<String> strBox2 = new Box<>("Banana");  
  
 System.*out*.println("strBox1: " + strBox1);  
 System.*out*.println("strBox2: " + strBox2);  
  
 comparisonResult = strBox1.compareTo(strBox2);  
 System.*out*.println("Сравнение strBox1 и strBox2: " +  
 (comparisonResult < 0 ? "strBox1 меньше" : comparisonResult > 0 ? "strBox1 больше" : "равны"));  
 }  
}

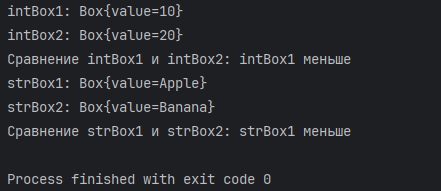
В методе compareTo учитывается, что значения внутри объектов Box могут быть null. Если хотя бы одно из значений не задано, метод выбрасывает исключение IllegalArgumentException.

Программа создаёт две пары объектов Box:

* Для хранения чисел типа Integer, где значения 10 и 20 сравниваются.
* Для хранения строк типа String, где сравниваются строки "Apple" и "Banana". Сравнение строк происходит в лексикографическом порядке.

Результаты работы метода compareTo обрабатываются и выводятся в читаемом виде с помощью тернарного оператора.

**Результат вывода программы:**



**Заключение:**

В ходе выполнения лабораторной работы я изучил основы использования обобщений в Java, научился создавать обобщённые классы и методы, а также использовать интерфейс Comparable<T> для сравнения объектов. Разработанный класс Box продемонстрировал универсальность и гибкость обобщений при работе с различными типами данных. Полученные знания позволят эффективно разрабатывать код, который может работать с разными типами данных, сохраняя безопасность типов и уменьшение дублирования кода.