



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ _____ «Информатика и системы управления»

КАФЕДРА _____ «Теоретическая информатика и компьютерные технологии»

Лабораторная работа № 4
по курсу «Языки и методы программирования»
«Реализация итераторов в языке Java»
Вариант 8

Студент группы ИУ9-22Б Павлов И. П.

Преподаватель Посевин Д. П.

Москва 2023

1 Цель работы

Изучение обобщённых итераторов и экземплярных вложенных классов языка Java.

2 Условие

Во время выполнения лабораторной работы требуется разработать на языке Java один из классов, перечисленных в таблицах 1 – 7. Класс должен реализовывать интерфейс `Iterable`. Объект разрабатываемого класса должен быть изменяемым, то есть в нём надо так или иначе предусмотреть возможность изменения внутреннего состояния.

Класс, представляющий множество дизъюнктов и частичное присваивание значений переменным. Дизъюнкт – это формула, представляющая собой дизъюнкцию булевских переменных.

Дизъюнкт удобно представлять множеством имён переменных. Частичное присваивание значений переменным может задаваться множеством имён переменных, имеющих значение `true`. Итератор должен выдавать дизъюнкты, принимающие значения `true`.

3 Реализация основного класса

```
public class Main {
    public static void main(String[] args) {
        Disjunct myDisjunct1 = new Disjunct();
        Disjunct myDisjunct2 = new Disjunct();

        myDisjunct1.addVar("a", false);
        myDisjunct1.addVar("b", true);

        myDisjunct2.addVar("c", false);
        myDisjunct2.addVar("d", false);

        DisjunctSet mySet = new DisjunctSet();
        mySet.addDisjunct(myDisjunct1);
        mySet.addDisjunct(myDisjunct2);

        for (Disjunct ignored : mySet) {
            System.out.println("Элемент проверен.");
        }

        System.out.println("Значение первого дизъюнкта: " + myDisjunct1.isTrue());
    }
}
```

```

        System.out.println("Значение второго дизъюнкта: " + myDisjunct2.isTrue());

        myDisjunct2.setVar("a", true);

        System.out.println("Второй дизъюнкт после изменения переменной: " + myDisjunct2.isTrue());
    }
}

```

4 Реализация Дизъюнкта

```

import java.util.HashSet;

public class Disjunct {
    private HashSet<String> trueVars = new HashSet<String>();
    private HashSet<String> vars = new HashSet<String>();

    public void addVar(String varName, boolean value) {
        if (value) {
            trueVars.add(varName);
        }
        vars.add(varName);
    }

    public void setVar(String varName, boolean value) {
        if (!value) {
            trueVars.remove(varName);
        } else {
            trueVars.add(varName);
        }
    }

    public boolean isTrue() {
        return trueVars.size() > 0;
    }

    @Override
    public String toString() {
        StringBuilder output = new StringBuilder("( ");
        for (String element : vars) {
            output.append(element).append(" ");
        }
        output.append(")");
        return output.toString();
    }
}

```

5 Реализация Множества Дизъюнктов

```

import java.util.HashSet;
import java.util.Iterator;

public class DisjunctSet implements Iterable<Disjunct> {
    public HashSet<Disjunct> disjuncts = new HashSet<Disjunct>();
}

```

```

private int size;

public DisjunctSet() {
    this.size = 0;
}

public void addDisjunct(Disjunct entity) {
    disjuncts.add(entity);
    this.size++;
}

@Override
public Iterator<Disjunct> iterator() {
    return new setIterator();
}

private class setIterator implements Iterator<Disjunct> {

    private int currentIndex = 0;
    private final Iterator<Disjunct> internalIterator = disjuncts.iterator();

    @Override
    public boolean hasNext() {
        return this.currentIndex < size;
    }

    @Override
    public Disjunct next() {
        Disjunct nextDisjunct = internalIterator.next();
        currentIndex++;
        if (nextDisjunct.isTrue()) {
            System.out.println(nextDisjunct);
        }
        return nextDisjunct;
    }
}
}
}

```

The screenshot shows an IDE with a Java file. The code includes a loop that iterates over a set of disjuncts, printing each one. It also shows a specific disjunct being set to true and then printed. The output window at the bottom shows the following text:

```

Run: Main
/usr/lib/jvm/java-19-openjdk/bin/java -javaagent:/home/darleet/.local/share/JetBrains/Toolbox/apps/IDEA-U/ch-0/223.8617.56/lib/ide...
Элемент проверен.
( a b )
Элемент проверен.
Значение первого дизъюнкта: true
Значение второго дизъюнкта: false
Второй дизъюнкт после изменения переменной: true
Process finished with exit code 0

```

Рис. 1: Вывод программы