TEHNICI AVANSATE DE PROGRAMARE
LABORATORUL 3

1. Expresii regulate:

Sunt instrumente folosite în procesarea textelor. Permit specificarea unor patternuri care pot fi regăsite în texte. Clasele utilizate în Java pentru lucrul cu expresii regulate sunt:

- **`java.util.regex.Matcher`** (obiectele de tip Matcher sunt obținute pornind de la un obiect de tip Pattern și vor permite compararea unui string cu o expresie regulată).
- **`java.util.regex.Pattern`** (obiecte de tip Pattern se obtin prin apelarea metodei statice compile și reprezintă o variantă compilată a unei expresii regulate)
- **`java.util.regex.PatternSyntaxException`**

EXP 1: Expresia (abc)+ = textul "abc" apare o dată sau de mai multe ori.
a{n} = a apare de exact n ori
a* = a apare de 0 sau mai multe ori
ab = a urmat de b
a | b = a sau b
. = orice caracter
[abc] = oricare dintre caracterele a, b, c
[^abc] = orice caracter cu exceptia lui a, b sau c
[a-zA-Z] = orice character intre a si z sau intre A si Z
[0 -9] = character numeric
^ = început de linie $ = sfârșit de linie
[a-z && [def]] = d,e sau f (intersecție)
\d prescurtare pentru [0-9]
\D prescurtare pentru [^0-9]
\s prescurtare pentru [\t\n\xOB\f\r]

EXP 2: Șirul "Exemplu" este recunoscut de urmatoarele expresii regulate:
Exemplu E.* [eE]xemplu [eE]x[aeiou][a-z]pl.*

EXP 3:

```
public class TestExpresiiRegulate {
	public static void main(String[] args) {
		Scanner sc = new Scanner(System.in);
		System.out.println("Dati textul");
		String text = sc.next();
		String expresie = text;
		while(expresie.equals("0") == false){
			Pattern p = Pattern.compile(expresie);
			Matcher m = p.matcher(text);
			while(m.find()) {
				System.out.println("match \""
						+ m.group()
						+ " intre " + m.start()
						+ "-" + (m.end() - 1));
			}
			System.out.println("Dati expresia");
			expresie = sc.next();
		}
	}
}
```

Metoda statică compile() a clasei Pattern returnează un obiect de tip Pattern pornind de la expresia regulată trimisă ca argument.
Metoda match() a clasei Pattern returnează un obiect de tip Matcher pornind de la input-ul (text) trimis ca argument. Obiectul Matcher va cauta în input (text) grupuri care să respecte pattern-ul stabilit (expresie).

EXE 1: Scrieți un program care să înlocuiască într-un text două sau mai multe spații consecutive cu linie nouă. Folosiți metoda replaceAll a clasei String.

EXE 2: Scrieți un program care să citească un fișier HTML și să scrie în alt fișier numărul linkurilor și adresele la care acestea fac referire. Presupuneți că likurile sunt introduse de tagul < a href = ”http:// ” >.

EXE 3: Scrieți un program care să parseze un fișiser HTML: pentru fiecare tag se va afișa denumirea, lista atributelor și a valorilor pe care le iau atributele.
Exemplu: tag: img
width: 200, 300
height: 234, 260
src: imagine.jpg, foto.gif

EXP 4: Program care scrie în fișierul out.txt

```
import java.io.*;
class TestScrieFisier{
	public static void main(String arg[]){
		PrintWriter pw;
		try{
			pw=new PrintWriter(new File("out.txt"));
		}
		catch(FileNotFoundException err){
			System.out.println("Probleme fisier iesire");
			pw=new PrintWriter(System.out);
		}
		pw.print("Prima linie in fisier\n");
		pw.print("A doua linie in fisier");
		pw.close();
	}
}
```

2. Clase generice

O clasă generică este o clasă care utilizează tipuri generice (care nu sunt cunoscute în momentul definirii clasei). Ca tipuri generice se pot utiliza doar tipurile referință.

EXP 5:

```
class ClasaGenerica<T> {
	T [] t;
	ClasaGenerica(T[] t){
		this.t = t;
	}
	public String toString(){
		String s = "";
		for(T t1: t){
			s +=t1.toString() + " ";
		}
		return s;
	}
}
```

```
class TestClasaGenerica{
      public static void main(String args[]){
            Integer [] v = {1,2,3,4,5};
            ClasaGenerica<Integer> obClasaGenerica =
                  new ClasaGenerica<Integer>(v);
            System.out.println(obClasaGenerica);
            System.out.print(obClasaGenerica.t[0] +
                        obClasaGenerica.t[1]);
      }
}
```

3. Colecții în Java

Prin intermediul colecțiilor în Java putem manipula diferite tipuri de date cum ar fi vecotrii, listele înlănțuite, tabelele de dispersie etc.
Tipul implicit al elementelor unei colecții este Object.
Arhitectura Java pentru colecții cuprinde:

- interfețe
- implementeri ale intefețelor
- algoritmi polimorfici ce pot fi utilizați pe diverse implementari ale colecțiilor (exp: sortare, căutare binară)

Interfață	Calsă
Set	HashSet
SortedSet	TreeSet
List	ArrayList, LinkedList Vector
Map	HashMap Hashtable
SortedMap	TreeMap

EXP 6: Algoritmi polimorfici, sortare O(nlogn)

```
import java.util.*;
public class Sort {
    public static void main(String[] args) {
        List<String> list = Arrays.asList(args);
        Collections.sort(list);
        System.out.println(list);
    }
}
```

EXP 7: Iteratori

```
import java.util.Iterator;
class C{
int i;
      C(int i){
            this.i = i;
      }
      public String toString(){
            return i + " ";
      }
}
```

```
class ColectieIterabila
      implements Iterator<C>, Iterable<C>{
      private C c1,c2,c3;
      int index;
      ColectieIterabila(){
            index = 0;
            c1 = new C(1);
            c2 = new C(2);
            c3 = new C(3);
      }
      public Iterator<C> iterator(){
            return this;
      }

      public boolean hasNext() {
            return false;
      }

      public C next() {
            return null;
      }
      public void remove() {

      }
}
class TestColectieIterabila{
      public static void main(String args[]){
            ColectieIterabila c = new ColectieIterabila();
            if (c.hasNext()){
                  System.out.print(c.next());
            }

            for(C c1:c){
                  System.out.print((C) c1);
            }
      }
}
```

EXE 4: Folosind clasa LinkedList evaluați o expresie aritmetica în formă poloneză inversă. Operatori: +, -, *, /. Operanzi numere întregi.
Exemplu: expresia 2 3 + 7 * corespunde (2 + 3) * 7

EXE 5: Scrieți un program care va cauta într-un director și în toate subdirectoarele acestuia, fișiere cu extensia java ale căror nume conțin o expresie regulată. Se va testa dacă acoladele se închid corect. Dacă într-un fișier acoladele nu se închid corect atunci se va arunca eroarea EroareParanteze.