

Compile

```
javac Program.java //compile
java program,Program
javac -Xlint:unchecked Program.java //check
warnings
```

File

```
package program;
public class Program{
    public static void main(String[] args){ ... }
```

Imports

```
import java.util.*;
    java.util.ArrayList;
    java.util.Scanner;
import java.io.*;
import java.math.*;
```

Modifiers

public - pristupne odvsadial
protected - v ramci package, triedy a podtried
private - v ramci triedy
(no modifier) - v ramci package
abstract - neimplementovana metoda / trieda
final
 var - nemenna
 method - neda sa Override
 class - neda sa dedit
static - methods sa tykaju celej triedy nie konkr. instance

Keywords

extends
 //vytvarenie podtried
 class Shape { ... }
 class Circle extends Shape { ... }
instanceof //boolean
 //je instanciou triedy
 if (x instanceof Shape){ ... }
@Override
 //pred prekryvanou metodu
super
 //volanie metody z nadtriedy v podtriede
 super.method();

this

//premenna **this** je referencia na konkr. objekt, na kt. bola metoda zavolana
 //napr.
 class Contact {
 private String email;
 public void setName(String name) {
 this.name = name; }
 }

I/O

```
//KONZOLA
System.out.print(String); / System.out.println(String);
System.out.format("%x",var);
//BUFFERED READER
BufferedReader input = new BufferedReader(new
    InputStreamReader(System.in));
String line = input.readLine() //null ak uz koniec
//SCANNER
Scanner input = new Scanner(System.in);
input.hasNext(); //boolean
input.next(); //dalsi znak
input.nextInt(); //convert. na int
//FILE input - BufferedReader/Scanner
// System.in -> "file.txt"
input.close()
//FILE output - PrintStream
PrintStream output = new PrintStream("file.txt");
output.print/println/format(String);
output.close();
```

Useful

ArrayList
 list.size() // # prvkov
 list.get(i) / ,add(x) / .set(i,x);
 ArrayList<type> newList = new ArrayList<type>-(list)
 //skopirovanie obsahu do novej instance
String
 text.length() //dlzka
 modText = text.substring(beg_i,end_i);
 //copy string od do
 text = String.format("%x",var); //format ako printf

Useful (cont)

```
char x = text.charAt(i);
```

Generic

```
class Node <T> {  
    private T data;  
    public Node(T data) {  
        this.data = data; }  
    public T getData() {  
        return data; }  
}
```



By [deleted]
cheatography.com/deleted-121063/

Not published yet.
Last updated 12th March, 2020.
Page 2 of 2.

Sponsored by **CrosswordCheats.com**
Learn to solve cryptic crosswords!
<http://crosswordcheats.com>