

ECM251 – Linguagens de Programação I

Aula 16 – L1/1, L2/1 e L3/1

Engenharia da Computação – 3ª série

Acesso a Arquivos de Texto em Java ***(L1/1, L2/1 e L3/1)***

2025

ECM251 – Linguagens de Programação I

Aula 16 – L1/1, L2/1 e L3/1

Horário

Terça-feira: 2 x 2 aulas/semana

- L1/1 (07h40min-09h20min): *Prof. Calvetti*;
- L1/2 (09h30min-11h10min): *Prof. Calvetti*;
- L2/1 (07h40min-09h20min): *Prof. Menezes*;
- L2/2 (11h20min-13h00min): *Prof. Calvetti*;
- L3/1 (09h30min-11h10min): *Prof. Evandro*;
- L3/2 (11h20min-13h00min): *Prof. Evandro*.

Acesso a Arquivo de Texto em Java

Tópico

- Acesso a Arquivo de Texto em Java

Acesso a Arquivo de Texto em Java

Definição



- Uma aplicação Java que necessite acessar arquivos e textos armazenados em memória retentiva, não volátil, envolve criar um programa Java que lê e/ou escreve em arquivos de texto;
- Isso é útil em muitos cenários, como processamento de dados, armazenamento de configurações ou *log*, manipulação de registros e muito mais.

ECM251 – Linguagens de Programação I

Acesso a Arquivo de Texto em Java

Exemplo 1



```
1 import java.io.File;
2 public class FileDemonstration
3 { // display information about file user specifies
4     public void analyzePath( String path )
5     { // create File object based on user input
6         File name = new File( path );
7         if( name.exists() ) // if name exists, output information about it
8         { // display file (or directory) information
9             System.out.printf( "%s\n%s\n%s\n%s\n%s\n%s\n%s\n%s\n%s\n%s\n",
10                 name.getName(), " exists",
11                 ( name.isFile() ? "is a file" : "is not a file" ),
12                 ( name.isDirectory() ? "is a directory" : "is not a directory" ),
13                 ( name.isAbsolute() ? "is absolute path" : "is not absolute path" ), "Last modified: ",
14                 name.lastModified(), "Length: ", name.length(),
15                 "Path: ", name.getPath(), "Absolute path: ",
16                 name.getAbsolutePath(), "Parent: ", name.getParent() );
17             if( name.isDirectory() ) // output directory listing
18             { String directory[] = name.list();
19                 System.out.println( "\n\nDirectory contents:\n" );
20                 for( String directoryName : directory )
21                     System.out.printf( "%s\n", directoryName );
22             } // end else
23         } // end outer if
24         else // not file or directory, output error message
25         { System.out.printf( "%s %s", path, "does not exist." );
26         } // end else
27     } // end method analyzePath
28 } // end class FileDemonstration
29
```

ECM251 – Linguagens de Programação I

Acesso a Arquivo de Texto em Java

Exemplo 1



```
1 import java.util.Scanner;
2 public class FileDemonstrationTest
3 {
4     public static void main( String args[] )
5     {
6         Scanner input = new Scanner( System.in );
7         FileDemonstration application = new FileDemonstration();
8
9         System.out.print( "Enter file or directory name here: " );
10        application.analyzePath( input.nextLine() );
11    } // end main
12 } // end class FileDemonstrationTest
13
```

ECM251 – Linguagens de Programação I

Acesso a Arquivo de Texto em Java

Exemplo 2



```
1 public class AccountRecord
2 { // A class that represents one record of information.
3     private int account;
4     private String firstName;
5     private String lastName;
6     private double balance;
7
8     // no-argument constructor calls other constructor with default values
9     public AccountRecord()
10    { this( 0, "", "", 0.0 ); // call four-argument constructor
11    } // end no-argument AccountRecord constructor
12
13    // initialize a record
14    public AccountRecord( int acct, String first, String last, double bal )
15    { setAccount( acct );
16      setFirstName( first );
17      setLastName( last );
18      setBalance( bal );
19    } // end four-argument AccountRecord constructor
20
21    public void setAccount( int acct ) // set account number
22    { account = acct;
23    } // end method setAccount
24
25    public int getAccount() // get account number
26    { return account;
27    } // end method getAccount
28
```

ECM251 – Linguagens de Programação I

Acesso a Arquivo de Texto em Java

Exemplo 2



```
29  public void setFirstName( String first )  // set first name
30  {  firstName = first;
31  }  // end method setFirstName
32
33  public String getFirstName()  // get first name
34  {  return firstName;
35  }  // end method getFirstName
36
37  public void setLastName( String last )  // set last name
38  {  lastName = last;
39  }  // end method setLastName
40
41  public String getLastName()  // get last name
42  {  return lastName;
43  }  // end method getLastName
44
45  public void setBalance( double bal )  // set balance
46  {  balance = bal;
47  }  // end method setBalance
48
49  public double getBalance()  // get balance
50  {  return balance;
51  }  // end method getBalance
52 }  // end class AccountRecord
53
```


ECM251 – Linguagens de Programação I

Acesso a Arquivo de Texto em Java

Exemplo 2



```
1 // Writing data to a text file with class Formatter.
2 import java.io.FileNotFoundException;
3 import java.lang.SecurityException;
4 import java.util.Formatter;
5 import java.util.FormatterClosedException;
6 import java.util.NoSuchElementException;
7 import java.util.Scanner;
8
9 public class CreateTextFile
10 { private Formatter output; // object used to output text to file
11   public void openFile() // enable user to open file
12   { try
13     { output = new Formatter( "clients.txt" );
14     } // end try
15     catch( SecurityException securityException )
16     { System.err.println( "You do not have write access to this file." );
17       System.exit( 1 );
18     } // end catch
19     catch( FileNotFoundException filesNotFoundException )
20     { System.err.println( "Error creating file." );
21       System.exit( 1 );
22     } // end catch
23   } // end method openFile
24 }
```

ECM251 – Linguagens de Programação I

Acesso a Arquivo de Texto em Java

Exemplo 2



```
25 public void addRecords() // add records to file
26 { // object to be written to file
27     AccountRecord record = new AccountRecord();
28     Scanner input = new Scanner( System.in );
29     System.out.printf( "%s\n%s\n%s\n%s\n\n",
30         "To terminate input, type the end-of-file indicator ",
31         "when you are prompted to enter input.",
32         "On UNIX/Linux/Mac OS X type <ctrl> d then press Enter",
33         "On Windows type <ctrl> z then press Enter or <ctrl> e" );
34     System.out.printf( "%s\n%s",
35         "Enter account number (> 0) <SPACE> first name <SPACE> last name <SPACE> balance <ENTER>",
36         "? " );
37     while ( input.hasNext() ) // loop until end-of-file indicator
38     { try // output values to file
39         { // retrieve data to be output
40             record.setAccount( input.nextInt() ); // read account number
41             record.setFirstName( input.next() ); // read first name
42             record.setLastName( input.next() ); // read last name
43             record.setBalance( input.nextDouble() ); // read balance
44             if ( record.getAccount() > 0 )
45             { // write new record
46                 output.format( "%d %s %s %.2f\n", record.getAccount(),
47                     record.getFirstName(), record.getLastName(),
48                     record.getBalance() );
49             } // end if
50             else
51             { System.out.println(
52                 "Account number must be greater than 0." );
53             } // end else
54         } // end try
```

ECM251 – Linguagens de Programação I

Acesso a Arquivo de Texto em Java

Exemplo 2



```
55     catch ( FormatterClosedException formatterClosedException )
56     { System.err.println( "Error writing to file." );
57       return;
58     } // end catch
59     catch ( NoSuchElementException elementException )
60     { System.err.println( "Invalid input. Please try again." );
61       input.nextLine(); // discard input so user can try again
62     } // end catch
63     System.out.printf( "%s %s\n%s", "Enter account number (>0) <SPACE>",
64       "first name <SPACE> last name <SPACE> balance <ENTER>", "? " );
65   } // end while
66 } // end method addRecords
67
68 public void closeFile() // close file
69 { if ( output != null )
70   output.close();
71 } // end method closeFile
72 } // end class CreateTextFile
73
```

ECM251 – Linguagens de Programação I

Acesso a Arquivo de Texto em Java

Exemplo 2



```
1 // Testing the CreateTextFile class.
2 public class CreateTextFileTest
3 {   public static void main( String args[] )
4     {   CreateTextFile application = new CreateTextFile();
5         application.openFile();
6         application.addRecords();
7         application.closeFile();
8     }   // end main
9 }   // end class CreateTextFileTest
10
```

ECM251 – Linguagens de Programação I

Acesso a Arquivo de Texto em Java

Exemplo 3



```
1 // This program reads a text file and displays each record.
2 import java.io.File;
3 import java.io.FileNotFoundException;
4 import java.lang.IllegalStateException;
5 import java.util.NoSuchElementException;
6 import java.util.Scanner;
7
8 public class ReadTextFile
9 { private Scanner input;
10   // enable user to open file
11   public void openFile()
12   { try
13     { input = new Scanner( new File( "clients.txt" ) );
14     } // end try
15     catch ( FileNotFoundException fileNotFoundException )
16     { System.err.println( "Error opening file." );
17       System.exit( 1 );
18     } // end catch
19   } // end method openFile
20
21   public void readRecords() // read record from file
22   { // object to be written to screen
23     AccountRecord record = new AccountRecord();
24     System.out.printf( "%-10s%-12s%-12s%10s\n", "Account",
25                       "First Name", "Last Name", "Balance" );
```

ECM251 – Linguagens de Programação I

Acesso a Arquivo de Texto em Java

Exemplo 3



```
26     try    // read records from file using Scanner object
27     { while( input.hasNext() )
28         { record.setAccount( input.nextInt() ); // read account number
29           record.setFirstName( input.next() ); // read first name
30           record.setLastName( input.next() ); // read last name
31           record.setBalance( input.nextDouble() ); // read balance
32           // display record contents
33           System.out.printf( "%-10d%-12s%-12s%10.2f\n",
34                               record.getAccount(), record.getFirstName(),
35                               record.getLastName(), record.getBalance() );
36         } // end while
37     } // end try
38     catch( NoSuchElementException elementException )
39     { System.err.println( "File improperly formed." );
40       input.close();
41       System.exit( 1 );
42     } // end catch
43     catch( IllegalStateException stateException )
44     { System.err.println( "Error reading from file." );
45       System.exit( 1 );
46     } // end catch
47 } // end method readRecords
48
49 public void closeFile() // close file and terminate application
50 { if( input != null )
51     input.close(); // close file
52 } // end method closeFile
53 } // end class ReadTextFile
54
```

ECM251 – Linguagens de Programação I

Acesso a Arquivo de Texto em Java

Exemplo 3



```
1 // This program test class ReadTextFile.
2 public class ReadTextFileTest
3 {   public static void main( String args[] )
4     {   ReadTextFile application = new ReadTextFile();
5         application.openFile();
6         application.readRecords();
7         application.closeFile();
8     }   // end main
9 }   // end class ReadTextFileTest
10
```


Exercícios



1. Desenvolver, utilizando o console como interface com o usuário, um programa que receba NOME e SENHA por pessoa cadastrada, digitados pelo usuário, e que, posteriormente, sejam, pelo aplicativo Java, gravados em um arquivo de texto.

Exercícios



2. Desenvolver, utilizando o console como interface com o usuário, um programa que receba um NOME e uma SENHA, digitados pelo usuário, para que sejam validados com o NOME e a SENHA lidos no arquivo de texto gerado pelo aplicativo do item 1, anterior.
 - Caso os Nomes e as Senhas sejam iguais, o aplicativo deverá apresentar a resposta “LOGIN REALIZADO COM SUCESSO”;
 - Caso as informações não coincidam, o aplicativo deverá apresentar a resposta “LOGIN E/OU SENHA ERRADO(S)”.

Exercícios



3. Desenvolver, utilizando Layouts como interface com o usuário, um programa que receba RA, NOME, SOBRENOME, P1, P2, P3 e P4 por pessoa cadastrada, digitados pelo usuário, e que, posteriormente, sejam, pelo aplicativo Java, gravados em um arquivo de texto.

Exercícios



4. Desenvolver, utilizando Layouts como interface com o usuário, um programa que receba um RA, digitado pelo usuário, para que sejam validados com os Ras lidos no arquivo de texto gerado pelo aplicativo do item 1, anterior.
 - Caso o RA digitado seja encontrado no arquivo texto, o aplicativo deverá apresentar o RA, o NOME e o SOBRENOME do aluno, bem como sua MÉDIA = $(P1 + P2 + P3 + P4)/4$;
 - Caso as informações não coincidam, o aplicativo deverá apresentar a resposta “RA NÃO CADASTRADO”.

ECM251 – Linguagens de Programação I

Aula 16 – L1/1, L2/1 e L3/1

Bibliografia Básica



- MILETTO, Evandro M.; BERTAGNOLLI, Silvia de Castro. Desenvolvimento de software II: introdução ao desenvolvimento web com HTML, CSS, javascript e PHP (Tekne). Porto Alegre: Bookman, 2014. E-book. Referência Minha Biblioteca:
<https://integrada.minhabiblioteca.com.br/#/books/9788582601969>
- WINDER, Russel; GRAHAM, Roberts. Desenvolvendo Software em Java, 3ª edição. Rio de Janeiro: LTC, 2009. E-book. Referência Minha Biblioteca:
<https://integrada.minhabiblioteca.com.br/#/books/978-85-216-1994-9>
- DEITEL, Paul; DEITEL, Harvey. Java: how to program early objects. Hoboken, N. J: Pearson, c2018. 1234 p. ISBN 9780134743356.

Continua...

Bibliografia Básica (continuação)



- HORSTMANN, Cay S; CORNELL, Gary. Core Java. SCHAFRANSKI, Carlos (Trad.), FURMANKIEWICZ, Edson (Trad.). 8. ed. São Paulo: Pearson, 2010. v. 1. 383 p. ISBN 9788576053576.
- LIANG, Y. Daniel. Introduction to Java: programming and data structures comprehensive version. 11. ed. New York: Pearson, c2015. 1210 p. ISBN 9780134670942.
- TURINI, Rodrigo. Desbravando Java e orientação a objetos: um guia para o iniciante da linguagem. São Paulo: Casa do Código, [2017]. 222 p. (Caelum).

ECM251 – Linguagens de Programação I

Aula 16 – L1/1, L2/1 e L3/1

Bibliografia Complementar



- HORSTMANN, Cay. Conceitos de Computação com Java. Porto Alegre: Bookman, 2009. E-book. Referência Minha Biblioteca:
<https://integrada.minhabiblioteca.com.br/#/books/9788577804078>
- MACHADO, Rodrigo P.; FRANCO, Márcia H. I.; BERTAGNOLLI, Silvia de Castro. Desenvolvimento de software III: programação de sistemas web orientada a objetos em java (Tekne). Porto Alegre: Bookman, 2016. E-book. Referência Minha Biblioteca:
<https://integrada.minhabiblioteca.com.br/#/books/9788582603710>
- BARRY, Paul. Use a cabeça! Python. Rio de Janeiro: Alta Books, 2012. 458 p.
ISBN 9788576087434.

Continua...

ECM251 – Linguagens de Programação I

Aula 16 – L1/1, L2/1 e L3/1

Bibliografia Complementar (continuação)



- LECHETA, Ricardo R. Web Services RESTful: aprenda a criar Web Services RESTful em Java na nuvem do Google. São Paulo: Novatec, c2015. 431 p.
ISBN 9788575224540.
- SILVA, Maurício Samy. JQuery: a biblioteca do programador. 3. ed. rev. e ampl. São Paulo: Novatec, 2014. 544 p.
ISBN 9788575223871.
- SUMMERFIELD, Mark. Programação em Python 3: uma introdução completa à linguagem Python. Rio de Janeiro: Alta Books, 2012. 506 p.
ISBN 9788576083849.

Continua...

ECM251 – Linguagens de Programação I

Aula 16 – L1/1, L2/1 e L3/1

Bibliografia Complementar (continuação)



- YING, Bai. Practical database programming with Java. New Jersey: John Wiley & Sons, c2011. 918 p.
- ZAKAS, Nicholas C. The principles of object-oriented JavaScript. San Francisco, CA: No Starch Press, c2014. 97 p. ISBN 9781593275402.

ECM251 – Linguagens de Programação I

Aula 16 – L1/1, L2/1 e L3/1

FIM

Engenharia da Computação – 3ª série

Acesso a Arquivo de Texto em Java **(L1/2, L2/2 e L3/2)**

2025

ECM251 – Linguagens de Programação I

Aula 16 – L1/2, L2/2 e L3/2

Horário

Terça-feira: 2 x 2 aulas/semana

- L1/1 (07h40min-09h20min): *Prof. Calvetti*;
- L1/2 (09h30min-11h10min): *Prof. Calvetti*;
- L2/1 (07h40min-09h20min): *Prof. Menezes*;
- L2/2 (11h20min-13h00min): *Prof. Calvetti*;
- L3/1 (09h30min-11h10min): *Prof. Evandro*;
- L3/2 (11h20min-13h00min): *Prof. Evandro*.

Exercícios



- Terminar, entregar e apresentar ao professor para avaliação, os exercícios propostos na aula de teoria, deste material.

ECM251 – Linguagens de Programação I

Aula 16 – L1/2, L2/2 e L3/2

Bibliografia (apoio)



- LOPES, ANITA. GARCIA, GUTO. Introdução à Programação: 500 algoritmos resolvidos. Rio de Janeiro: Elsevier, 2002.
- DEITEL, P. DEITEL, H. Java: como programar. 8 Ed. São Paulo: Prentice-Hall (Pearson), 2010;
- BARNES, David J.; KÖLLING, Michael. Programação orientada a objetos com Java: uma introdução prática usando o BlueJ . 4. ed. São Paulo: Pearson Prentice Hall, 2009.

ECM251 – Linguagens de Programação I

Aula 16 – L1/2, L2/2 e L3/2

FIM