|  | ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ  ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΤΗΛΕΜΑΤΙΚΗΣ |
| --- | --- |

Κουβαράς Μαρίνος

Κρασάκης Γεώργιος

1η Εργασία στο μάθημα **«Ανάπτυξη Λογισμικού ΙΙ»**

Ταύρος, 23 Μαΐου 2024

Περιεχόμενα

[**Περίληψη 3**](#_gjdgxs)

[**Δομή 4**](#_5lsat0x4xdc4)

[**Main.java 5**](#_9on2042qn09m)

[**Student.java 5**](#_6vazxtm0rhea)

[**InputMenu.java 7**](#_ctel8vj87lmv)

[**MyApp.java 10**](#_l9hs8vlz806u)

[**MyUtils.java 18**](#_6r2wwkhe1xlc)

[**Ενδεικτικές εκτελέσεις (screenshots): 22**](#_svbmyvz438ky)

[Αρχικό μενού επιλογών 22](#_1sfuszroxhh5)

[Επιλογή 1 22](#_eohtmett7vjz)

[Επιλογή 2 23](#_nzcqja5cisfh)

[Επιλογή 3 23](#_ndmy92txwbov)

[Επιλογή 4 24](#_yw03eow3rzju)

[Επιλογή 5 24](#_708hrenfuwdh)

[Επιλογή 6 24](#_4qjdrwzfzv2r)

[Επιλογή 7 25](#_egq4c1mhaaj)

[Επιλογή 8 25](#_30yh19okggqg)

[Επιλογή 9 25](#_cwngalphn7h2)

[**Γενικά Σχόλια/Παρατηρήσεις 25**](#_30kjkxhk7qt9)

# 

# 

# Περίληψη

Στόχος της εργασίας είναι να δημιουργηθεί μια εφαρμογή Java η οποία θα αποθηκεύει φοιτητές σε μια λίστα και θα επιτρέπει διάφορες λειτουργίες πάνω σε αυτήν τη λίστα. Συγκεκριμένα, όταν ξεκινάει θα εμφανίζει ένα μενού με τις εξής επιλογές:

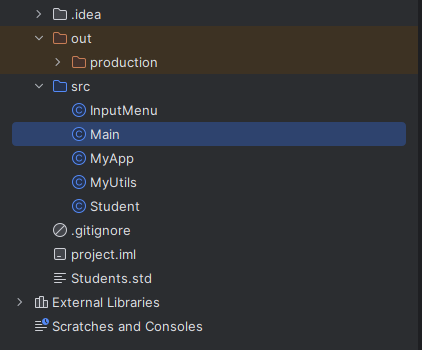
* 1. View all students: Θα εκτυπώνει όλη τη λίστα με τους φοιτητές.
* 2. Add student: Θα προσθέτει έναν φοιτητή στη λίστα.
* 3. Delete a student: Θα διαγράφει έναν φοιτητή από τη λίστα με βάση το id του.
* 4. Modify a student: Θα επιτρέπει την αλλαγή μιας μεταβλητής του φοιτητή, πχ βαθμολογία, όνομα, τμήμα. ΟΧΙ το id.
* 5. Print a student: Θα αναζητεί και θα εκτυπώνει έναν φοιτητή με βάση το id του.
* 6. Sort list of students: Θα ταξινομεί τη λίστα με βάση τη βαθμολογία των φοιτητών (φθίνουσα σειρά).
* 7. Save list to a file: Θα αποθηκεύει τη λίστα των φοιτητών σε αρχείο.
* 8. Load list from a file: Θα φορτώνει τη λίστα φοιτητών από αρχείο.
* 9. Exit: Θα τερματίζει την εφαρμογή.

# 

# 

# Δομή

Η δομή του προγράμματος φαίνεται παρακάτω.



* [**Main**](#_9on2042qn09m): Η κλάση αυτή είναι η κύρια κλάση που εκκινεί την εφαρμογή μας
* [**Student**](#_6vazxtm0rhea): Η κλάση αυτή δημιουργεί και περιγράφει την οντότητα του μαθητή όπως ζητείται στην εκφώνηση
* [**InputMenu**](#_ctel8vj87lmv): Η κλάση αυτή δημιουργεί το κύριο μενού επιλογών και διάδρασης
* [**MyApp**](#_l9hs8vlz806u): Η κλάση αυτή περιλαμβάνει την κύρια λογική της εφαρμογής μας
* [**MyUtils**](#_6r2wwkhe1xlc): Η κλάση αυτή δημιουργήθηκε επιβοηθητικά και περιλαμβάνει επαναχρησιμοποιήσιμες μεθόδους - εργαλεία για την εκτέλεση ελέγχων και διεργασιών.

### 

### 

### 

### 

### 

### 

### Main.java

Ο κώδικας που δημιουργήθηκε μαζί με τα σχόλια είναι:

| import java.util.ArrayList;  import java.util.Scanner;  public class Main {  public static void main(String[] args) {  ArrayList<Student> students= new ArrayList<>();  Scanner scanner = new Scanner(System.*in*);  new InputMenu(scanner, students);  scanner.close();  }  } |
| --- |

### Student.java

Ο κώδικας που δημιουργήθηκε μαζί με τα σχόλια είναι:

| public class Student implements java.io.Serializable {  private int id;  private String stName;  private String stSurname;  private String stDepartment;  private double stGrade;  public Student(int id, String name, String surname, String department, double grade) {  this.id = id;  this.stName = name;  this.stSurname = surname;  this.stDepartment = department;  this.stGrade = grade;  }  */\*\**  *\* GETTERS*  *\*/*  public int getId() {  return id;  }  public String getName() {  return stName;  }  public String getSurname() {  return stSurname;  }  public String getDepartment() {  return stDepartment;  }  public double getGrade() {  return stGrade;  }  */\*\**  *\* SETTERS*  *\*/*  public void setId(int id) {  this.id = id;  }  public void setName(String name) {  this.stName = name;  }  public void setSurname(String surname) {  this.stSurname = surname;  }  public void setDepartment(String department) {  this.stDepartment = department;  }  public void setGrade(double grade) {  this.stGrade = grade;  }  @Override  public String toString() {  return "Student{ " +  "id=" + id +  ", Name='" + stName + '\'' +  ", Surname='" + stSurname + '\'' +  ", Department='" + stDepartment + '\'' +  ", Grade=" + stGrade +  " }";  }  } |
| --- |

### InputMenu.java

Ο κώδικας που δημιουργήθηκε μαζί με τα σχόλια είναι:

| import java.util.ArrayList;  import java.util.Scanner;  public class InputMenu {  */\*\**  *\* Display Menu*  *\*/*  public void display\_menu() {  System.*out*.println(  """  #######################  1) View all students  2) Add student  3) Delete a student  4) Modify a student  5) Print a student  6) Sort list of students  7) Save list to a file  8) Load list from a file  9) Exit  #######################  SELECTION:"""  );  }  */\*\**  *\* Input selection*  *\*/*  public InputMenu(Scanner scanner, ArrayList<Student> students) {  int selection; //option selection  do {  display\_menu();  selection = scanner.nextInt(); //get selection number  switch (selection) {  case 1:  System.*out*.println("You selected 'View all students' ");  MyApp.*viewAllStudents*(students);  break;  case 2:  System.*out*.println("You selected 'Add student' ");  MyApp.*addStudent*(students);  break;  case 3:  System.*out*.println("You selected 'Delete a student' ");  MyApp.*deleteStudent*(students);  break;  case 4:  System.*out*.println("You selected 'Modify a student' ");  MyApp.*modifyStudent*(students);  break;  case 5:  System.*out*.println("You selected 'Print a student' ");  MyApp.*printStudent*(students);  break;  case 6:  System.*out*.println("You selected 'Sort list of students' ");  MyApp.*sortByGrade*(students);  break;  case 7:  System.*out*.println("You selected 'Save list to a file' ");  MyApp.*saveToFile*(students);  break;  case 8:  System.*out*.println("You selected 'Load list from a file' ");  students = MyApp.*loadFromFile*(students);  // Check if loadedStudents is not null and contains data  if (students != null && !students.isEmpty()) {  for (Student student : students) {  System.*out*.println(student);  }  }  break;  case 9:  System.*out*.println("You selected 'Exit' ");  //method();  break;  default:  System.*err*.println("Unrecognized option");  break;  }  } while (selection != 9);  scanner.close();  }  } |
| --- |

### MyApp.java

Ο κώδικας που δημιουργήθηκε μαζί με τα σχόλια είναι:

| import java.io.\*;  import java.util.\*;  public class MyApp {  */\*\**  *\* Selection 1.View all students*  *\* @param students arrayList*  *\*/*  public static void viewAllStudents(ArrayList<Student> students) {  if (students.isEmpty()) {  System.*out*.println(MyUtils.*getMessage*("emptyList"));  }else {  System.*out*.println(students); //return all students  }  }  */\*\**  *\* Selection 2.Add student*  *\* @param students arrayList*  *\*/*  public static void addStudent(ArrayList<Student> students) {  Scanner myEntry = new Scanner(System.*in*);  System.*out*.println("Please add ID, Name, Surname, Department, Grade: PRESS ENTER AFTER EACH ITEM INSERTION");  while (true) {  try {  // ID  MyUtils.*printInfo*("ID");  String input = myEntry.nextLine();  input = MyUtils.*validateInputID*("ID: ", input, "invalidInput");  int id = Integer.*parseInt*(input); //read input  if (MyUtils.*isIdExists*(students, id)) {  continue;  }  // NAME  MyUtils.*printInfo*("NAME");  String name = myEntry.nextLine(); //read input  name = MyUtils.*validateInputChars*("NAME: ", name, "invalidCharacters");  // SURNAME  MyUtils.*printInfo*("SURNAME");  String surname = myEntry.nextLine();  surname = MyUtils.*validateInputChars*("SURNAME: ", surname, "invalidCharacters");  // DEPARTMENT  MyUtils.*printInfo*("DEPARTMENT");  String department = myEntry.nextLine();  department = MyUtils.*validateInputChars*("DEPARTMENT: ", department, "invalidCharacters");  // GRADE  double grade;  while (true) {  try {  MyUtils.*printInfo*("GRADE");  grade = myEntry.nextDouble();  grade = MyUtils.*validateInputGrade*("GRADE: ", grade, "invalidGrade");  break;  } catch (InputMismatchException e) {  System.*out*.println(MyUtils.*getMessage*("invalidInput"));  myEntry.nextLine();  }  }  Student myStudent = new Student(id, name, surname, department, grade); //create new Student  students.add(myStudent); //add new Student  System.*out*.println("New Student added successfully");  } catch (InputMismatchException e) {  System.*out*.println(e.getMessage());  myEntry.nextLine();  }  break;  }  }  */\*\**  *\* Selection 3.Delete a student*  *\* @param students arrayList*  *\*/*  public static void deleteStudent(ArrayList<Student> students) {  Scanner myEntry = new Scanner(System.*in*);  if (students.isEmpty()) {  System.*out*.println(MyUtils.*getMessage*("emptyList"));  } else {  try {  System.*out*.println("Select student's id to delete");  // ID  String input = myEntry.nextLine();  input = MyUtils.*validateInputID*("ID: ", input, "invalidInput");  int selection = Integer.*parseInt*(input);  //Find the student with the matching ID  Student toDeleteStudent = null;  for (Student student : students ) {  if (student.getId() == selection ) {  toDeleteStudent = student;  break;  }  }  if (toDeleteStudent != null) {  students.remove(toDeleteStudent);  System.*out*.println(toDeleteStudent + " removed.");  } else {  System.*out*.println("Student not found");  }  } catch (IndexOutOfBoundsException e) {  System.*out*.println("ID out of range, please try again.");  } catch (InputMismatchException e) {  System.*out*.println("Invalid input format. Please enter a valid number next time.");  }  }  }  */\*\**  *\* Selection 4.Modify a student*  *\* @param students arrayList*  *\*/*  public static void modifyStudent(ArrayList<Student> students) {  Scanner myEntry = new Scanner(System.*in*);  if (students.isEmpty()) {  System.*out*.println(MyUtils.*getMessage*("emptyList"));  } else {  while (true) {  try {  System.*out*.println("Select student's id to modify");  // ID  String input = myEntry.nextLine();  input = MyUtils.*validateInputID*("ID: ", input, "invalidInput");  int selection = Integer.*parseInt*(input);  //Find the student with the matching ID  Student toModifyStudent = null;  for (Student student : students ) {  if (student.getId() == selection ) {  toModifyStudent = student;  break;  }  }  if (toModifyStudent != null) {  System.*out*.println("You selected: "+ toModifyStudent);  System.*out*.println("Please modify Name, Surname, Department, Grade: PRESS ENTER AFTER EACH ITEM INSERTION");  // NAME  MyUtils.*printInfo*("NAME");  String name = myEntry.nextLine(); //read input  name = MyUtils.*validateInputChars*("NAME: ", name, "invalidCharacters");  MyUtils.*printInfo*("SURNAME");  String surname = myEntry.nextLine();  surname = MyUtils.*validateInputChars*("SURNAME: ", surname, "invalidCharacters");  MyUtils.*printInfo*("DEPARTMENT");  String department = myEntry.nextLine();  department = MyUtils.*validateInputChars*("DEPARTMENT: ", department, "invalidCharacters");  // GRADE  double grade;  while (true) {  try {  MyUtils.*printInfo*("GRADE");  grade = myEntry.nextDouble();  grade = MyUtils.*validateInputGrade*("GRADE: ", grade, "invalidGrade");  break;  } catch (InputMismatchException e) {  System.*out*.println(MyUtils.*getMessage*("invalidInput"));  myEntry.nextLine();  }  }  toModifyStudent.setName(name);  toModifyStudent.setSurname(surname);  toModifyStudent.setDepartment(department);  toModifyStudent.setGrade(grade);  System.*out*.println("Student modified successfully");  break;  } else {  System.*out*.println("Student not found");  break;  }  } catch (IndexOutOfBoundsException e) {  System.*out*.println("ID out of range");  }  }  }  }  */\*\**  *\* Selection 5.Print student*  *\* @param students arrayList*  *\*/*  public static void printStudent(ArrayList<Student> students) {  Scanner myEntry = new Scanner(System.*in*);  if (students.isEmpty()) {  System.*out*.println(MyUtils.*getMessage*("emptyList"));  } else {  try {  System.*out*.println("Select student's id to show");  // ID  String input = myEntry.nextLine();  input = MyUtils.*validateInputID*("ID: ", input, "invalidInput");  int selection = Integer.*parseInt*(input);  //Find the student with the matching ID  Student toPrintStudent = null;  for (Student student : students ) {  if (student.getId() == selection ) {  toPrintStudent = student;  break;  }  }  if (toPrintStudent != null) {  System.*out*.println(toPrintStudent);  } else {  System.*out*.println("Student not found");  }  } catch (IndexOutOfBoundsException e) {  System.*out*.println("ID out of range");  }  }  }  */\*\**  *\* Selection 6.Sort by grade*  *\* @param students arrayList*  *\*/*  public static void sortByGrade(ArrayList<Student> students) {  if (students.isEmpty()) {  System.*out*.println(MyUtils.*getMessage*("emptyList"));  } else {  Comparator<Student> gradeComparator = Comparator.*comparingDouble*(Student::getGrade);  students.sort(gradeComparator.reversed());  // Print each student's information separately  for (Student student : students) {  System.*out*.println(student);  }  }  }  */\*\**  *\* Selection 7.Save to file*  *\* @param students arrayList*  *\*/*  public static void saveToFile(ArrayList<Student> students) {  Scanner myEntry = new Scanner(System.*in*);  System.*out*.print("Enter the file name to save (e.g. students): ");  String fileName = myEntry.nextLine();  try {  FileOutputStream fileOut = new FileOutputStream(fileName);  ObjectOutputStream out = new ObjectOutputStream(fileOut);  out.writeObject(students);  out.close();  fileOut.close();  System.*out*.printf("Serialized data is saved in " + fileName + '\n');  } catch (IOException e) {  System.*out*.println("Error saving data to file: " + e.getMessage());  }  }  */\*\**  *\* Selection 8.Load from file*  *\* @return arrayList of type Student*  *\*/*  public static ArrayList<Student> loadFromFile(ArrayList<Student> students) {  Scanner myEntry = new Scanner(System.*in*);  System.*out*.print("Enter the file name to load (e.g. students): ");  String fileName = myEntry.nextLine();  //myEntry.close();  try {  FileInputStream fileIn = new FileInputStream(fileName);  ObjectInputStream in = new ObjectInputStream(fileIn);  ArrayList<Student> loadedStudents = (ArrayList<Student>) in.readObject();  in.close();  fileIn.close();  students.addAll(loadedStudents);  System.*out*.println("Load successfully. Here are the results:");  } catch (IOException e) {  System.*out*.println("Error reading from file: " + e.getMessage());  } catch (ClassNotFoundException e) {  System.*out*.println("Error loading data: " + e.getMessage());  }  return students;  }  } |
| --- |

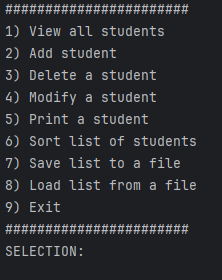
### MyUtils.java

Ο κώδικας που δημιουργήθηκε μαζί με τα σχόλια είναι:

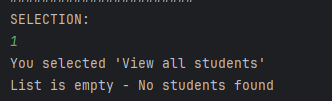
| import java.util.ArrayList;  import java.util.Scanner;  import java.util.InputMismatchException;  */\*\**  *\* MyUtils handles different input validations along with some messages*  *\*/*  public class MyUtils {  // Method for input prompting ID, NAME etc.  public static void printInfo(String message) {  System.*out*.println(message+ ": ");  }  // method for checking unique id  public static boolean isIdExists(ArrayList<Student> students, int id) {  //iterate over the arrayList  for (Student student : students) {  if (student.getId() == id) {  System.*out*.println("ID already exists. Please try a different ID.");  return true;  }  }  return false;  }  */\*\**  *\**  *\* @param messageType of type string*  *\* @return a String with the appropriate message*  *\*/*  public static String getMessage(String messageType) {  return switch (messageType) {  case "emptyList" -> "List is empty - No students found"; // message for empty list  case "invalidInput" -> "Invalid input format. Please enter a valid number."; // message for invalid number input  case "invalidCharacters"-> "Invalid input format. Please enter only alphabetic characters."; // message for invalid characters input  case "invalidGrade" -> "Invalid grade. Grade must be between 0 and 10."; // message for grade value out of permitted range  // Add more cases for other message types as needed  default -> "Unknown message type";  };  }  */\*\**  *\**  *\* @param value of the input field ID*  *\* @param input the input we want to check*  *\* @param messageType the type of message we want to handle the error*  *\* @return the inserted id*  *\*/*  public static String validateInputID(String value, String input, String messageType) {  Scanner scanner = new Scanner(System.*in*);  while (!input.matches("\\d+")) { //prevent spaces insertion  System.*out*.println(MyUtils.*getMessage*(messageType));  System.*out*.print(value);  input = scanner.nextLine();  }  return input;  }  */\*\**  *\**  *\* @param value input field NAME, SURNAME, DEPARTMENT*  *\* @param chars the input we want to check*  *\* @param messageType the type of message we want to handle the error*  *\* @return the validated input*  *\*/*  public static String validateInputChars(String value,String chars, String messageType) {  Scanner scanner = new Scanner(System.*in*);  String validatedChars = chars; // Variable to hold the validated input  while (!validatedChars.matches("[a-zA-Z]+")) {  System.*out*.println(MyUtils.*getMessage*(messageType));  System.*out*.print(value);  validatedChars = scanner.nextLine();  }  return validatedChars; // Return the validated input  }  */\*\**  *\**  *\* @param value input grade*  *\* @param grade the value we want to check*  *\* @param messageType the type of message we want to handle the error*  *\* @return the validated value*  *\*/*  public static double validateInputGrade(String value, double grade, String messageType) {  Scanner scanner = new Scanner(System.*in*);  while (grade < 0 || grade > 10.0) {  try {  System.*out*.println(MyUtils.*getMessage*(messageType));  System.*out*.print(value);  grade = scanner.nextDouble();  // Check if the entered grade is within the valid range  if (grade >= 0 && grade <= 10.0) {  break; // Exit the loop if the grade is valid  } else {  System.*out*.println("Invalid grade. Grade must be between 0 and 10.");  }  } catch (InputMismatchException e) {  // Clear the invalid input from the scanner buffer  scanner.next();  System.*out*.println("Invalid input. Please enter a numeric value.");  }  }  return grade;  }  } |
| --- |

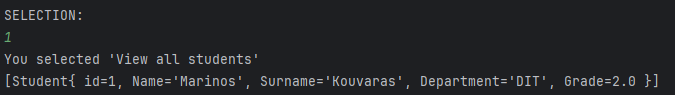
## **Ενδεικτικές εκτελέσεις (screenshots):**

### Αρχικό μενού επιλογών



### Επιλογή 1





### 

### 

### 

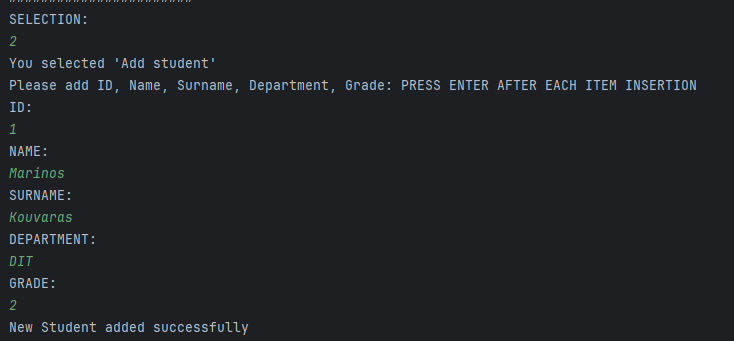
### 

### 

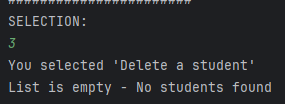
### 

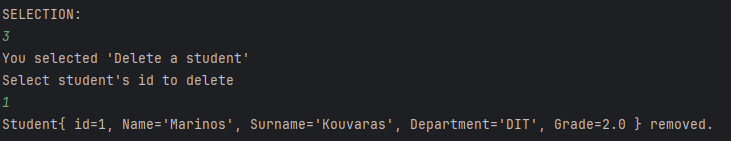
### 

### Επιλογή 2



### Επιλογή 3

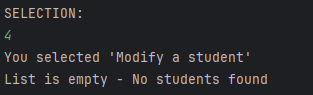


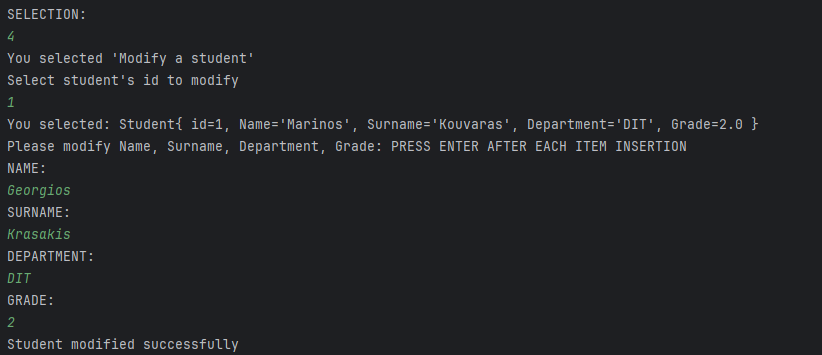


### 

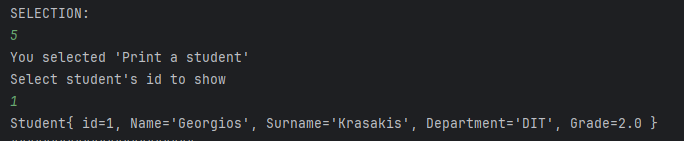
### 

### Επιλογή 4

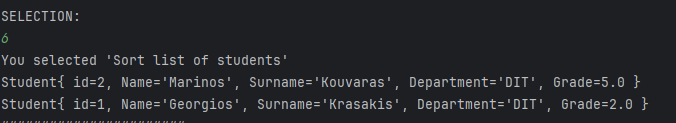




### Επιλογή 5

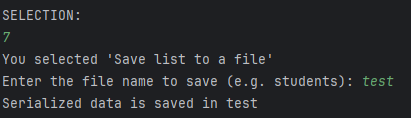


### Επιλογή 6

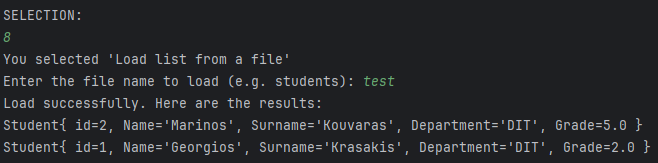


### 

### Επιλογή 7



### Επιλογή 8



### Επιλογή 9

