-------------MAIN-----------------

Student S1 = **new** Student("John", "1209");

Student S2 = **new** Student("Nick"); S1.setName("John Brown");

Course C1 = **new** Course("Java", 5); S2.setId("1210");

(Κάθε φοιτητης ένα μαθημα) (Βαζουμε τα αντοιστιχα set

S1.setMyCourse(C1); στην STUDENT)

S2.setMyCourse(C1); (ουσιαστικα στο αντικειμενο του μαθητη

S1.printInfo(); βαζουμε το αντικ του μαθηματος)

S2.printInfo();

--------- COURSE---------

**private** String name;

**private** **int** credits;

**public** Course(String text, **int** number)

name = text;

credits = number;

**public** String getName()

**return** name;

**public** **int** getCredits()

**return** credits;

---------STUDENT---------

**private** String name;

**private** String id;

**private** Course myCourse; (ΠΡΟΣΟΧΗ)

**public** Student(String text1, String text2) {

name = text1;

id = text2;

**public** Student(String text1) { (2 ΚΑΤΑΣΚΕΥΑΣΤΕΣ,ΙΔΙΟ ΟΝΟΜΑ)

name = text1;

id = "9999";

(εδώ γινετε η συσχετηση)

**public** **void** setMyCourse(Course c) {

myCourse = c;

**public** String getName() {

**return** name;

**public** String getID() {

**return** id;

**public** **void** printInfo() {

Systet.out.println("Name: " + name);

System.out.println("AM: " + id);

System.out.println("Course: " + myCourse.getName());

System.out.println("Credits: " + myCourse.getCredits());

--------------MAIN----------------

Student S1 = **new** Student("John", "1209");

Student S2 = **new** Student("Nick", "1709");

Course C1 = **new** Course("Math101");

C1.addStudent(S1);

C1.addStudent(S2); (εδώ ένα μαθημα εχει πολλους μαθητες)

C1.printStudentsInfo();

------------COURSE--------------

**private** String name;

**private** ArrayList<Student> enrolledStudents = **new** ArrayList<Student>();

**public** Course(String text) {

name = text;

**public** **void** addStudent(Student s) {

enrolledStudents.add(s);

**public** **void** printStudentsInfo() {

**for**(**int** i=0; i<enrolledStudents.size(); i++)

enrolledStudents.get(i).printInfo();

-------Η STUDENT εχει κατασκευαστη και μια απλη printInfo()-----------

-------------MAIN-------------

Student S1 = **new** Student("John", "1209");

Student S2 = **new** Student("Nick", "17173");

Course C1 = **new** Course("Java");

Course C2 = **new** Course("PHP");

Course C3 = **new** Course("maths");

S1.addCourse(C1);

S2.addCourse(C1);

C1.printInformation();

S1.printData();

--------------STUDENT----------------

**private** String name;

**private** String id;

**private** ArrayList<Course> courses = **new** ArrayList<Course>();

**public** Student(String text1, String text2) {

name = text1;

id = text2;

**public** **void** addCourse(Course aCourse) {

courses.add(aCourse);

aCourse.addStudent(**this**); (η αμφιδρομη επιτυχαν με αυτην την εντολη)

**public** String getName() {

**return** name;

**public** String getID() {

**return** id;

**public** **void** printData() {

System.***out***.println("Name: " + name);

System.***out***.println("AM: " + id);

**for**(Course c: courses) {

System.***out***.println(c.getName());

-------------------COURSE----------------------

**private** String name;

**private** ArrayList<Student> students = **new** ArrayList<Student>();

**public** Course(String text) {

name = text;

**public** String getName() {

**return** name;

**public** **void** addStudent(Student aStudent) {

students.add(aStudent);

**public** **void** printInformation() {

System.***out***.println("Course Name: " + name);

System.***out***.println("has student: " );

**for**(Student s: students)

System.***out***.println(s.getName());

Ουσιαστικα στην πρωτη συσχετηση βαζουμε στο αντικειμενο S1 το αντικειμενο C2 και όταν κανουμε την αμφιδρομη βαζουμε στο C2 το S1…

ΚΛΗΡΟΝΟΜΙΚΟΤΗΤΑ

----------------MAIN----------------

BankAccount BA1 = **new** BankAccount();

BA1.deposit(1500);

BankAccount BA2 = **new** BankAccount();

BA2.deposit(5000);

SavingsAccount SA1 = **new** SavingsAccount();

SA1.deposit(2000);

SA1.setRate(5);

BA2.printInformation();

SA1.printInformation();

-----------------BankAccount--------------

**private** **double** balance;

**protected** **double** newBalance; private double newBalance;

(και με public δουλευει αλλα όχι σωστο) ++ public double getBalance(){

**public** BankAccount(){ return newBalance;

balance = 50; }

**public** **void** deposit(**double** amount) {

newBalance = balance + amount;

**public** **void** printInformation() {

System.***out***.println("This is a normal Bank Account");

System.***out***.println("With balance: " + newBalance);

--------------SavingsAccount---------------

... Extends BankAccount

**private** **double** rate;

**public** **void** setRate(**double** number) {

rate = number;

(overrides the method of the superclass)

**public** **void** printInformation() {

System.***out***.println("This is a SAVINGS ACCOUNT");

System.***out***.println("with balance: " + newBalance);

getBalance() αντι για newBalance

System.***out***.println("and interest rate: " + rate);

Αντι να κανουμε επιπλεον μια συναρτηση και να καλουμε αυτην την συναρτηση πολύ απλα κανουμε την μεταβλητη μας protected…

Αποτελεσματα: This is a normal Bank Account

With balance: 5050.0

This is a SAVINGS ACCOUNT

with balance: 2050.0

and interest rate: 5.0

-------------MAIN------------

Employee ref = **new** Employee("John", 850);

ref.printStatement();

ref = **new** AdministrativeEmployee("Bob", 800, 10, 15);

ref.printStatement();

-------------Employee------------

**private** String name;

**private** **double** salary;

**public** Employee(String text, **double** amount) {

name = text;

salary = amount;

**public** **void** printStatement() {

System.***out***.println("Employee: " + name + " with salary: " + salary);

--------------AdministrativeEmployee-------------

...Extends Employee

**private** **int** overtimeHours;

**private** **double** overtimeWage;

**public** AdministrativeEmployee(String text, **double** amount, **int** hours, **double** wage) {

**super**(text, amount);

overtimeHours = hours;

overtimeWage = wage;

(εδώ ναι μεν κανω override αλλα ταυτοχρονα καλο και την αρχικη)

**public** **void** printStatement() {

**super**.printStatement();

System.***out***.println("and extra income: " + (overtimeHours \* overtimeWage));

ΑΠΟΤΕΛΕΣΜΑΤΑ: Employee: John with salary: 850.0

Employee: Bob with salary: 800.0

and extra income: 150.0

------------------ΜΑΙΝ------------------

BankAccount BA1 = **new** BankAccount(1000);

SavingsAccount SA1 = **new** SavingsAccount(500);

CheckingAccount CA1 = **new** CheckingAccount(1200);

Bank LoseBank = **new** Bank(); (ειναι σαν να φτιαχνει ενα κενο

Αντικειμενο ώστε να μπορει να καλει

LoseBank.add(BA1); Την add)

LoseBank.add(SA1);

LoseBank.add(CA1);

LoseBank.printAllData();

------------------BANK---------------------

**private** ArrayList<BankAccount> accounts = **new** ArrayList<BankAccount>();

**public** **void** add(BankAccount account) {

accounts.add(account);

**public** **void** printAllData() {

**for**(BankAccount account : accounts) { (ΠΟΛΥΜΟΡΦΙΣΜΟΣ)

account.printData();

-------------------BankAccount------------------

**protected** **double** balance;

**public** BankAccount(**double** amount) {

balance = amount;

**public** **void** printData() {

System.***out***.println("Normal BankAccount");

System.***out***.println("with balance: " + balance);

-----------------CheckingAccount------------------

... Extends BankAccount

**private** **int** transactionsCounter;

**public** CheckingAccount(**double** amount) {

**super**(amount);

transactionsCounter = 0;

**public** **void** printData() {

System.***out***.println("CheckingAccount");

System.***out***.println("with balance:" + balance+"free transactions:3");

---------------SavingsAccount-------------------

... Extends BankAccount

**private** **double** interestRate;

**public** SavingsAccount(**double** amount) {

**super**(amount);

**public** **void** printData() {

System.***out***.println("SavingsAccount");

System.***out***.println("with balance"+balance+"and rate"+ interestRate);

ΑΠΛΗ ΑΣΚΗΣΗ 1

------------MAIN---------------

Company c1 = **new** Company("EpsilonNet");

c1.addEmployee();

c1.addEmployee();

c1.printEmployeeRecords();

------------Company-------------

**private** String name;

**private** ArrayList<Employee> employees = **new** ArrayList<Employee>();

**public** Company(String name) {

**this**.name = name;

**public** **void** addEmployee() {

String answer = JOptionPane.*showInputDialog*("What type of Emplo....

**double** employeeSalary = Double.*parseDouble*(employeeSalaryText);

**double** bonus = 0;

**if**(answer.equalsIgnoreCase("T")) {

String bonusText = JOptionPane.*showInputDialog*("Enter….

bonus = Double.*parseDouble*(bonusText);

**if**(answer.equalsIgnoreCase("E"))

employees.add(**new** Employee(employeeName, employeeSalary));

**else** {

TechEmployee te = **new** TechEmployee(employeeName, employeeSalary);

te.setBonus(bonus);

employees.add(te);

**public** **void** printEmployeeRecords() {

System.***out***.println("Company: " + name);

**double** totalIncome = 0;

**for**(Employee employee: employees) {

totalIncome += employee.printIncome();

}

System.***out***.println("Total Income: " + totalIncome);

-------------------Employee--------------

**protected** String name;

**protected** **double** salary;

**public** Employee(String name, **double** salary) {

**public** **double** printIncome() {

**double** monthlyIncome = salary \* 0.8;

System.***out***.println("Monthly income for " + name +

" is: " + monthlyIncome);

**return** monthlyIncome;

--------------TechEmployee------------

... Extends Employee

**private** **double** bonus;

**public** TechEmployee(String name, **double** salary) {

**super**(name, salary);

**public** **void** setBonus(**double** bonus) {

**this**.bonus = bonus;

**public** **double** printIncome() {

**double** monthlyIncome = salary \* 0.7 + bonus;

System.***out***.println("Monthly income for " + name +

" is: " + monthlyIncome);

**return** monthlyIncome;

ΑΠΛΗ ΑΣΚΗΣΗ 2

---------------MAIN----------------

ArrayList<Student> students = **new** ArrayList<Student>();

DataInput input = **new** DataInput();

input.read(students);

**for**(Student student : students) {

student.printInfo();

------------Datalnput-------------

**public** **void** read(ArrayList<Student> students) {

**boolean** more = **true**;

**while**(more) {

String studentType = JOptionPane.*showInputDialog*("What type of student? (A: Student, B: MScStudent)");

String name = JOptionPane.*showInputDialog*("Enter Name");

Student student;

**if**(studentType.equals("A"))

student = **new** Student(name);

**else** {

String supervisorName =

JOptionPane.*showInputDialog*("Enter Supervisor");

student = **new** MScStudent(name, supervisorName);

}

students.add(student); (ΠΡΟΣΟΧΗΗΗ)

String answer = JOptionPane.*showInputDialog*("More? (yes/no)");

**if**(answer.equals("no"))

more = **false**;

---------------Student-----------------

**private** String name;

**public** Student(String name) {

**this**.name = name;

**public** **void** printInfo() {

System.***out***.println("Name: " + name);

------------MscStudent-----------

Extends Student…

**private** String supervisor;

**public** MScStudent(String name, String supervisor) {

**super**(name);

**this**.supervisor = supervisor;

**public** **void** printInfo() {

**super**.printInfo();

System.***out***.println("Supervisor: " + supervisor);

ABSTRACT

----------------MAIN-------------

ArrayList<Student> students = **new** ArrayList<Student>();

students.add(**new** UnderGraduate("Babis", 7.78));

students.add(**new** PhDStudent("Takis", "Troika & Anergia"));

**for**(Student student: students)

student.printInfo();

--------------Student-------------

**public** **abstract** **class** Student {

**private** String name;

**public** Student(String name) {

**this**.name = name;

**public** String getName() {

**return** name;

**public** **abstract** **void** printInfo();

--------------UnderGraduate-------------

**public** **class** UnderGraduate **extends** Student {

**private** **double** GPA;

**public** UnderGraduate(String name, **double** gPA) {

**super**(name);

GPA = gPA;

**public** **void** printInfo() {

System.***out***.println("Undergraduate Student");

System.***out***.println("Name: " + getName());

System.***out***.println("GPA: " + GPA);

--------------PhDStudent-------------

**public** **class** PhDStudent **extends** Student {

**private** String thesis;

**public** PhDStudent(String name, String thesis) {

**super**(name);

**this**.thesis = thesis;

**public** **void** printInfo() {

System.***out***.println("PhD Student");

System.***out***.println("Name: " + getName());

System.***out***.println("Thesis: " + thesis);

INTERFACE

DataSet ds = **new** DataSet();

BankAccount BA1 = **new** BankAccount(1500);

BankAccount BA2 = **new** BankAccount(2000);

BankAccount BA3 = **new** BankAccount(7000);

ds.add(BA1);

ds.add(BA2);

ds.add(BA3);

System.***out***.println("Average is: " + ds.getAverage());

System.***out***.println("Maximum is: " + ds.getMaximum().getMeasure());

System.***out***.println("Minimum is: " + ds.getMinimum().getMeasure());

(στα 2 τελευταια επιστρεφει αντικειμενο γιαυτο βαζουμε το getMeasure())

---------------DataSet---------------

**private** **double** sum;

**private** Measurable max;

**private** Measurable min;

**private** **int** counter;

**public** DataSet() {

sum = 0;

counter = 0;

max = **null**;

min = **null**;

**public** **void** add(Measurable item) {

sum = sum + item.getMeasure();

**if**(counter == 0) {

max = item;

min = item;

**else** **if**(item.getMeasure() > max.getMeasure())

max = item;

**else** **if**(item.getMeasure() < min.getMeasure())

min = item;

counter++;

**public** **double** getAverage() {

**if**(counter == 0)

**return** 0;

**else**

**return** sum/counter;

**public** Measurable getMaximum() {

**return** max;

**public** Measurable getMinimum() {

**return** min;

---------------Measurable---------------

**public** **interface** Measurable {

**double** getMeasure();

---------------BankAccount---------------

**public** **class** BankAccount **implements** Measurable {

**private** **double** balance;

**public** BankAccount(**double** amount) {

balance = amount;

**public** **double** getBalance() {

**return** balance;

**public** **double** getMeasure() {

**return** getBalance();

--------------- Coin ---------------

**public** **class** Coin **implements** Measurable {

**private** **double** value;

**public** Coin(**double** amount) {

value = amount;

**public** **double** getValue() {

**return** value;

**public** **double** getMeasure() {

**return** value;

ΓΡΑΦΙΚΑ

centralPanel = **new** JPanel(**new** BorderLayout());

containerPanel = **new** JPanel(**new** GridLayout(3, 2));

containerPanel.add(codeField);

containerPanel.add(destinationField)΄

containerPanel.add(weightField);

containerPanel.add(powerField);

containerPanel.add(bulkButton);

containerPanel.add(refridgeratorButton);

centralPanel.add(containerPanel, BorderLayout.***CENTER***);

shipList = **new** JList();

DefaultListModel listModel = **new** DefaultListModel();

**for**(Ship ship: ships) {

listModel.addElement(ship.getName());

}

shipList.setModel(listModel);

centralPanel.add(shipList, BorderLayout.***NORTH***);

Πανω είναι η λιστα και από κατω εχει 2στηλες 3γραμμες με textfield+button

ΔΟΜΕΣ

LinkedList<String> list = **new** LinkedList<String>();

list.add("John");

list.add("Mary");

Iterator<String> iter = list.iterator();

**1) while**(iter.hasNext()) {

String name = iter.next();

System.***out***.println(name);

**2) for**(String name: list)

System.***out***.println(name);

Collection<BankAccount> set = **new** TreeSet<BankAccount>();

BankAccount BA1 = **new** BankAccount("001", 1500, "Papadopoulos");

BankAccount BA2 = **new** BankAccount("002", 2500, "Nikolaou");

BankAccount BA3 = **new** BankAccount("003", 1000, "Petrou");

set.add(BA1);...

**class** BankAccount **implements** Comparable {

**private** String id;

**private** **double** balance;

**private** String holderName;

**public** **int** compareTo(Object other) {

BankAccount otherAccount = (BankAccount)other;

**if**(**this**.balance < otherAccount.balance)

**return** -1;

**else** **if**(**this**.balance > otherAccount.balance)

**return** 1;

**else ΠΡΟΣΟΧΗ!! Συκρινω με βαση το μεσεο**

**return** 0;

}

**public** BankAccount(String id, **double** balance, String holderName) {

**this**.id = id;

**this**.balance = balance;

**this**.holderName = holderName;

**public** String getId() {

**return** id;

**public** **double** getBalance() {

**return** balance;

**public** String getHolderName() {

**return** holderName;

------------------------------------------------------------------

Collection<String> inputs = **new** ArrayList<String>();

New HashSet...

inputs.add("John");

inputs.add("Mary"); -----ΓΕΝΙΚΑΑΑΑ------

----------------------------------------------------------------------

HashSet<String> set = **new** HashSet<String>();

set.add("John");

set.add("Mary");

**for**(String name: set)

System.***out***.println(name);

**if**(set1.containsAll(set2))

System.***out***.println("set2 is a subset of s1");

**else**

System.***out***.println("set2 is NOT a subset of s1");

set1.addAll(set2);

System.***out***.println("---Union----");

System.***out***.println(set1);

set1.retainAll(set2);

System.***out***.println("---InterSection----"); APOT: [D, E, F]

System.***out***.println(set1);

System.***out***.println("---Set Difference----");

set1.add("X");

System.***out***.println(set1); [D, E, F, X]

System.***out***.println(set2); [D, E, F]

set1.removeAll(set2);

System.***out***.println(set1); [X]

---MAIN---

BankAccount BA1 = **new** BankAccount(1500, "001");

BankAccount BA2 = **new** BankAccount(1700, "001");

BankAccount BA3 = **new** BankAccount(1500, "002");

HashSet<BankAccount> hashSet = **new** HashSet<BankAccount>();

hashSet.add(BA1);

hashSet.add(BA2);

hashSet.add(BA3);

**for**(BankAccount account: hashSet)

System.***out***.println(account.getCode() + ", "

+ account.getBalance());

---BankAccount---

**public** BankAccount(**double** balance, String code) {

**this**.balance = balance;

**this**.code = code;

**public** **double** getBalance() {

**return** balance;

**public** String getCode() {

**return** code;

**public** **int** hashCode() { (ΠΡΟΣΟΧΗ)

**return** code.hashCode();

**public** **boolean** equals(Object other) {

BankAccount otherAccount = (BankAccount)other; (ΠΡΟΣΟΧΗ)

**if**(**this**.code.equals(otherAccount.code))

**return** **true**;

**else ¨ελεγχει¨ με βαση τον κωδικο οποτε**

**return** **false**; τα αποτελεσματα θα είναι 1500 ‘001’

1700 ‘002’

---------------------------------------

(μια περιπτωση με λιστα)

ArrayList<String> names = Collections.*list*(model.elements());

Collections.*sort*(names);

model.clear();

**for**(String name: names)

model.addElement(name);

-------------------------------------

1) HashMap<String, String> map = **new** HashMap<String, String>();

map.put("Nick", "19878547");

map.put("Mary", "25345345"); (τυχαια σειρα στα αποτελεσματα)

map.put("Helen", "35235");

map.put("Mike", "43534523");

**for**(String name: map.keySet()) {

String telephone = map.get(name);

System.***out***.println(name + " : " + telephone);

2) ArrayList<String> JohnBooks = **new** ArrayList<String>();

JohnBooks.add("Catcher in the Rye");

JohnBooks.add("Asterix");

ArrayList<String> MaryBooks = **new** ArrayList<String>();

MaryBooks.add("Nefelim");

MaryBooks.add("Giati mas psekazoun?");

HashMap<String, ArrayList<String>> map = **new** HashMap<String, ArrayList<String>>();

map.put("John", JohnBooks);

map.put("Mary", MaryBooks);

**for**(String name: map.keySet()) {

ArrayList<String> borrowedBooks = map.get(name);

System.***out***.println(name + " has borrowed the following Books: ");

**for**(String bookTitle : borrowedBooks)

System.***out***.println(bookTitle);

----------------------------------------

ΕΡΓΑΣΙΑ 1(Tic-Tac-Toe)

Scanner input = **new** Scanner(System.***in***);

**char**[][] board = **new** **char**[3][3];

Random dice = **new** Random();

**for** (**int** i=0; i<3; i++)

**for** (**int** j=0; j<3; j++)

board[i][j] = ' ';

**boolean** flag=**true**,flag2=**false**

thesi = input.nextLine();

**if**(!"A1".equals(thesi) & !"A2".equals(thesi) &...)

**if** (board[gr][st]=='X' || board[gr][st]=='O')

**do**

{

gr = dice.nextInt(3) ;

st = dice.nextInt(3) ;

}**while**(board[gr][st] == 'X' || board[gr][st] == 'O');

ΕΡΓΑΣΙΑ 3

**MAIN: new** GUI\_1();

Airport: κλασικο κατασκευαστη + getters

(επικαλυπτω την tostring)

**public** String toString() {

**return** (city + ", " + kwd + " airport");

CentralRegistry:

**private** **static** ArrayList<Airport> *airports* = **new** ArrayList<Airport>();

**private** **static** ArrayList<Flight> *flights* = **new** ArrayList<Flight>();

// ΜΕΘΟΔΟΣ εισαγωγης αεροπλανων

**public** **static** **void** addAirport(Airport x) { (ομοια για flights)

*airports*.add(x);

// ArrayList getters

**public** **static** ArrayList<Airport> getAirports() { (ομοια για flights)

**return** *airports*;

// ΜΕΘΟΔΟΣ ευρεσης αεροδρομιου σε μια πολη

**public** **static** Airport getAirport(String cityName) {

**int** thesi = -1;

**for**(**int** i=0; i<*getAirports*().size(); i++) {

**if** (cityName.equals(*getAirports*().get(i).getCity()))

thesi = i;

}

**if** (thesi != -1)

**return** *getAirports*().get(thesi);

**else** **return** **null**;

// ΜΕΘΟΔΟΣ ευρεσης στοιχει πτησης συνδεει απευθειας το αεροδρομιο a και b

**public** **static** String getDirectFlightsDetails(Airport a, Airport b) {

String alfari8m1 = "";

**int** counter = 0;

**for**(**int** i=0; i<*getFlights*().size(); i++)

**if**(*getFlights*().get(i).getAirportA().equals(a) && ...{

counter++;

alfari8m1+= "["+counter+"]"+ *getFlights*().get(i)+"\n";

}

**if**("".equals(alfari8m1))

**return** "there is not direct flights...";

**else**

**return** alfari8m1;

}

Flight: ομοια με Airport

GUI\_1:

**public** **class** GUI\_1 **extends** JFrame {

**private** JTextField eisag;

**private** JPanel panel;

**private** JButton button;

**public** GUI\_1() {

panel = **new** JPanel(); (Το 13 είναι ποσο κενο θελω μετα)

eisag = **new** JTextField("give a city name", 13);

button = **new** JButton("Find");

panel.add(eisag);

panel.add(button);

ButtonListener1 listener = **new** ButtonListener1();

button.addActionListener(listener);

**this**.setContentPane(panel);

**this**.setVisible(**true**);

**this**.setSize(350, 150);

**this**.setTitle("Find Aiport");

**this**.setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

}

// ΑΚΡΟΑΤΗΣ κουμπιου FIND

**class** ButtonListener1 **implements** ActionListener{

**public** **void** actionPerformed(ActionEvent e) {

**if**(e.getSource() == button) { (trim: ελεγχει τα κενα)

String cityName = eisag.getText().trim();

Airport c = CentralRegistry.*getAirport*(cityName);

**if** (c == **null**)

JOptionPane.*showMessageDialog*(**null**,cityName + " does not have an airport");

**else** { (message error!!)

**new** GUI\_2(c);

setVisible(**false**);(για να κλινει το παραθ)

GUI\_2:

**private** JPanel panel1;

**private** JTextField eksag1;

**private** JTextField eksag2; ...

**private** Airport airport;

**private** JList airList;

**private** DefaultListModel model1;

**private** JScrollPane scorllPane1;

**private** JPanel panel2;

**private** JButton findButton;

**private** JTextField eisagPoli;

**private** JPanel panel3;

**private** JTextArea detailsList;

**private** JTextArea indirectList;

**private** JPanel panel4;

**private** JButton backButton;

**private** JPanel centralPanel;

**public** GUI\_2(Airport aAirport) {

//---------------- BHMA 1 ----------------//

airport = aAirport;

panel1 = **new** JPanel();

panel1.setBorder(**new** EtchedBorder(EtchedBorder.***LOWERED***,**null**,**null**));

eksag1 = **new** JTextField(airport.getName(), 8);

eksag2 = **new** JTextField(airport.getKwd(), 8);

eksag3 = **new** JTextField(airport.getCity(), 8);

eksag4 = **new** JTextField(airport.getCountry(), 8);

airList = **new** JList();

model1 = **new** DefaultListModel();

airList.setModel(model1);

scorllPane1 = **new** JScrollPane(airList);

scorllPane1.setPreferredSize(**new** Dimension(80, 80));

panel1.add(eksag1);

panel1.add(eksag2);

panel1.add(eksag3);

panel1.add(eksag4);

panel1.add(scorllPane1);

Collection<String> set1 = **new** HashSet<String>();

**for**(**int** i=0; i<CentralRegistry.*getFlights*().size(); i++) {

Airport x = CentralRegistry.*getFlights*().get(i).getAirportA();

Airport y = CentralRegistry.*getFlights*().get(i).getAirportB();

**if**(x.equals(airport) || y.equals(airport))

set1.add(CentralRegistry.*getFlights*().get(i).getName\_air());

}

Collection<String> set2 = **new** TreeSet<String>(set1);

**for**(String nameAir: set2)

model1.addElement(nameAir);

//------------------------ ΒΗΜΑ 2 ----------------------------//

eisagPoli = **new** JTextField("give a destination",13);

findButton = **new** JButton("Find Flights");

ButtonListener2 listener = **new** ButtonListener2();

findButton.addActionListener(listener);

panel2 = **new** JPanel();

panel2.add(eisagPoli); panel2.add(findButton);

//---------------- ΒΗΜΑ 3-4 ----------------//

panel3 = **new** JPanel();

detailsList = **new** JTextArea();

indirectList = **new** JTextArea(); (περιγραμα)

detailsList.setBorder(**new** EtchedBorder(EtchedBorder.***LOWERED***, **null**, **null**));

indirectList.setBorder(**new** EtchedBorder(EtchedBorder.***LOWERED***, **null**, **null**));

GridLayout grid = **new** GridLayout(1,2);

panel3.setLayout(grid);

panel3.add(detailsList);

panel3.add(indirectList);

//----------- ΤΕΛΕΥΤΑΙΑ ΒΗΜΑΤΑ ------------//

panel4 = **new** JPanel();

backButton = **new** JButton("Back to Search Screen");

panel4.add(backButton);

backButton.addActionListener(listener);

centralPanel = **new** JPanel();

BoxLayout border = **new** BoxLayout(centralPanel, BoxLayout.***PAGE\_AXIS***);

centralPanel.setLayout(border);

centralPanel.add(panel1);

centralPanel.add(panel2);

centralPanel.add(panel3);

centralPanel.add(panel4);

**this**.setContentPane(centralPanel);

**this**.setVisible(**true**);

**this**.setSize(700, 500);

**this**.setTitle("Airport Page");

**this**.setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

}

// ΑΚΡΟΑΤΗΣ κουμπιων

**class** ButtonListener2 **implements** ActionListener{

**public** **void** actionPerformed(ActionEvent e) {

**if**(e.getSource() == backButton){

**new** GUI\_1();

setVisible(**false**);

}

**else** **if**(e.getSource() == findButton)

{

String cityName = eisagPoli.getText();

**if**(cityName.equals(airport.getCity()))

JOptionPane.*showMessageDialog*(**null**," Arrival and departure ... the same!");

**else** {

String s1 = "DIRECT FLIGHTS DETAILS \n" + CentralRegistry.*getDirectFlightsDetails*(airport, CentralRegistry.*getAirport*(cityName));

detailsList.setText(s1);

String s2 = "INDIRECT FLIGHTS through... \n" + CentralRegistry.*getInDirectFlightsDetails*(airport, CentralRegistry.*getAirport*(cityName));

indirectList.setText(s2);

ΕΡΓΑΣΙΑ 4

(Μεσα στο else της κλασης ακροατης ButtonListener2)

(δινω στο αρχειο το ονομα που θελω αλλιως μπορουσα πολύ απλα "MAKIS.txt")

File namesFile = **new** File(airport.getCity() + "To" + cityName + ".txt");

**try** {

FileWriter writer = **new** FileWriter(namesFile);

writer.write("CITY: " + airport.getCity() + ", " + airport.getCountry() + "\n");

writer.write("Aiport: " + airport.getName() + " (" + airport.getKwd() + ")" + "\n" + "\n");

writer.write("DESTINATION: " + cityName + "\n" + "\n");

writer.write("DIRECT FLIGHTS DETAILS: \n" + CentralRegistry.*getDirectFlightsDetails*(airport, CentralRegistry.*getAirport*(cityName)) + "\n" + "\n");

writer.write("INDIRECT FLIGHTS through... \n" + CentralRegistry.*getInDirectFlightsDetails*(airport, CentralRegistry.*getAirport*(cityName)));

writer.close();

} **catch** (IOException e1) {

e1.printStackTrace();

} (ΕΓΓΡΑΦΗ ΣΕ ΑΡΧΕΙΟ)

(πιο απλο παραδειγμα)

Arraylist…………. Jack,mary,bob,Helen...

File namesFile = new File(“makis.txt”);

**try** {

FileWriter writer = **new** FileWriter(namesFile);

for(String name: names)

writer.write(name);

writer.write(System.lineSeparator());

writer.close();

} **catch** (IOException e1) {

e1.printStackTrace();

}

ΑΝΑΓΝΩΣΗ ΑΠΟ ΑΡΧΕΙΟ

**private** JFileChooser fc = **new** JFileChooser();

**private** JPanel panel = **new** JPanel();

**private** JButton button = **new** JButton("Open File");

**public** GUI() {

panel.add(button);

//this.getContentPane().add(panel);

**this**.setContentPane(panel);

button.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent e) {

**int** returnVal = fc.showOpenDialog(panel);

**if**(returnVal == JFileChooser.***OPEN\_DIALOG***) {

File fιle = fc.getSelectedFile();

**try** {

BufferedReader reader = **new** BufferedReader(**new** FileReader(file));

String line = reader.readLine();

**while**(line != **null**)

{

System.***out***.println(line);

line = reader.readLine();

}

reader.close();

} **catch** (FileNotFoundException exc) {

exc.printStackTrace();

}

**catch**(IOException exc) {

exc.printStackTrace();

}

ΕΓΓΡΑΦΗ ΚΑΙ ΑΝΑΓΝΩΣΗ ΣΕ 2ΑΔΙΚΟ + συσχετ

Car car1 = **new** Car("BMW");

Car car2 = **new** Car("Porsche");

Car car3 = **new** Car("Maserati");

Employee e1 = **new** Employee("John");

Employee e2 = **new** Employee("Mary");

Employee e3 = **new** Employee("Bob");

e1.setCar(car2);

e3.setCar(car1);

e2.setCar(car3);

ArrayList<Employee> employees = **new** ArrayList<Employee>();

employees.add(e1);

employees.add(e2);

employees.add(e3);

File file = **new** File("employee.ser"); (αντι για αυτην θα μπορουσα στο file

**try** { να εβαζα το // "employee.ser")

FileOutputStream fileOut = **new** FileOutputStream(file);

ObjectOutputStream out = **new** ObjectOutputStream(fileOut);

out.writeObject(employees);

out.close();

fileOut.close();

System.***out***.println("Employees have been written");

} **catch** (FileNotFoundException exc1) {

exc1.printStackTrace();

} **catch** (IOException exc1) {

exc1.printStackTrace();

}

**try** {

FileInputStream fileIn = **new** FileInputStream("employee.ser");

ObjectInputStream in = **new** ObjectInputStream(fileIn);

ArrayList<Employee> retrievedEmployees = (ArrayList<Employee>)in.readObject();

in.close();

fileIn.close();

System.***out***.println("Employees retrieved: ");

**for**(Employee e: retrievedEmployees) {

System.***out***.println(e.getName() + " and has a: " + e.getCar().getBrand());

}

} **catch** (FileNotFoundException e11) {

e11.printStackTrace();

} **catch** (IOException e11) {

e11.printStackTrace();

} **catch** (ClassNotFoundException e)

e.printStackTrace();

**public** **class** Employee **implements** Serializable {

**private** String name;

**private** Car myCar;

**public** **void** setCar(Car c) {

myCar = c;

**public** Car getCar() {

**return** myCar;

**public** Employee(String name) {

**this**.name = name;

**public** String getName() {

**return** name;

**public** **class** Car **implements** Serializable {

**private** String brand;

**public** String getBrand() { το αρχειο Serialization3

**return** brand κανει το ιδιο αλλα με κλασεις

**public** Car(String brand) {

**this**.brand = brand; Το 4 είναι με συσχετηση