Exam Project
Introduction to Programming
August 22nd 2017

Details about how to submit the project

Deliver your project as a zip file named fall-exam-2017-USER-STUDENTID.zip (substitute 'USER' with your name and 'STUDENTID' with your student id number). This zip file is obtained by using the **export** command in Eclipse. This can be done by first choosing the File>Export menu command. Then select as destination in the 'General' folder the **Archive File** type. Be sure to select then the right project, the zip format and check the radio button "Create directory structure for files" (default). Include in the zip file all the files (bin, sources, resources), the .classpath and .project files. You can consult this page https://mcuoneclipse.com/2012/05/07/exporting-and-importing-projects-in-eclipse/ for an example. I must be able to **import** your Eclipse project, compile and execute the program without any additional library that is not already included in the standard Java API.

Double check that one of your colleagues using an OS different from yours and with Eclipse can successfully import, load and execute your project without any problem. Do not put in your code absolute references to files that cannot be found when running your project on another computer (my computer).

Note that if I cannot run your project or the application does not find the required files (see the requirements) you will **fail the exam**.

Send an email to fricci@unibz.it with this zip file before the deadline: **September 5th 2017 at noon.**

Student Code Ethics

Students are expected to maintain the highest standards of academic integrity. Work that is not of the student's own creation will receive no credit. Remember that you cannot give or receive unauthorized aid on any assignment, quiz, or exam. A student cannot use the ideas of another and declare it as his or her own. Students are required to properly cite the original source of the ideas and information used in his or her work.

Application requirements

Implement a swing- awt-based **eTourism** application where you can perform the tasks described in the following. The application will basically let the user to browse a catalogue of tourist destinations in South Tyrol, search for a particular destination, select one of them for obtaining more information.

The implemented application must be displayed in the main window that you will create. **Do not create an Applet.**

There must be a main window menu containing the following menu items:

- *Exit:* kills the application
- *About:* pops up a dialogue that tells something about the developer of the application (you) and some information about the application: number of classes, number of methods, total number of lines of code.

eTourism Functions

- There are least 20 destinations in your application; for instance, Meran, Bozen, Vinshgau, etc.
- Destinations' data are stored in a file, which is read by the application at the start up.
- Collect a set of test destinations data, and build your destinations file. Use the following web site https://www.suedtirol.info/en/regions
- A destination must be described by (at least): title, slogan, description, contact information (name, phone, email, web-site), average star rating, number of ratings (from different users of the application), and a picture (put all the pictures in a specific directory).
- Initially (home screen) show a map of South Tyrol and indicate where the various destinations are located. This can be implemented in a simple way, with a picture, or in a more advanced way (not required) with a figure with mouse-over effect, so that when you move the mouse over the tourist areas the names of the destinations are shown.
- The user can ask to see the list of all the destinations or search for a specific one (string search). The destinations (described in the file) are then visualized in a list in the main window.
- The destinations' list can then be sorted by the user alphabetically or according to average rating.
- Implement a search function that enables the user to search for a destination whose description, slogan or title contain a given string (do not match case – search for all strings that contain the input string). Put the search form in the home screen of the application and in the screen where the list of the destinations is shown.
- The user, from the list screen, can select a destination (click a button) that will be then illustrated with all the available details in another screen of the main window (e.g., full description and larger picture).
- A destination can be rated by the user in five stars rating scale. For each destination, the system keeps updated the average star ratings given by all the users. In order to compute the average rating, you must store all the ratings given to a destination, in

different usage sessions of the application, in the file containing the destinations' descriptions.

- The user should be able to compose an email in a form, in order to request more information about a selected destination. After the user confirms the submission of the email composition form the application should inform that he/she will receive the requested information. This occurs only if all the required form fields where provided by the user. Otherwise the user should be able to complete the form or discard it. Note that your application does not need to send any real email.