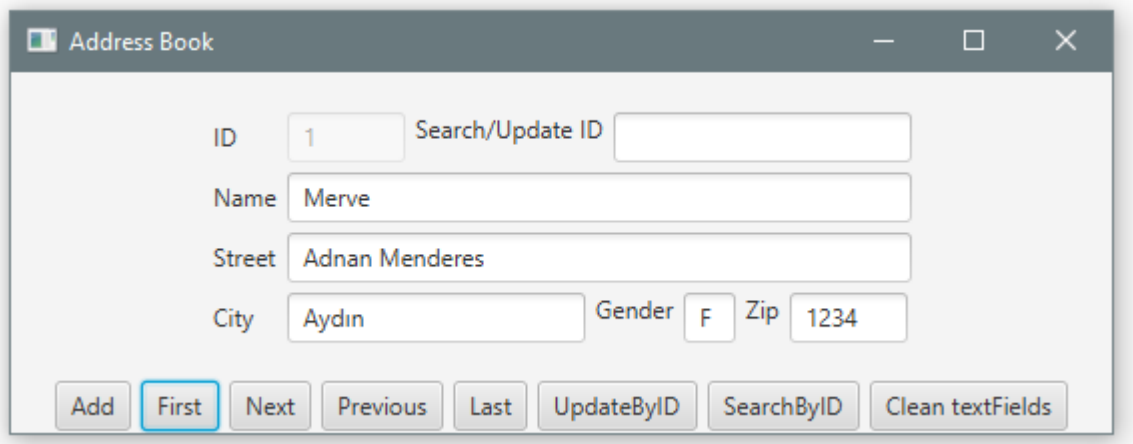
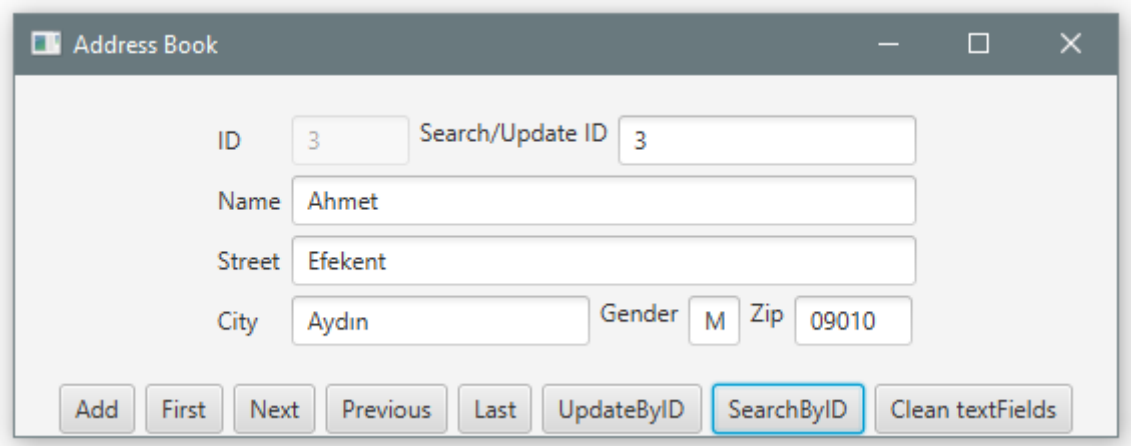
ADNAN MENDERES UNIVERSITY
CSE 203 Object-Oriented Programming

Project Address Book

- You should submit **whole java project file in zipped format** and **project report,** named project file as ***studentNo_NameSurname_Project***.
- Do your homeworks in **ECLIPSE IDE**.
- Do not use Turkish Characters(ç,ğ,ı,ö,ş,ü) for naming project, methods, classes.
- Late submissions are not allowed.
- Group working is allowed. Indicate all project members in the report. Each member of project should upload project file and report to Classroom.
- Copy homework will be evaluated as 0.
- **DO NOT upload a screenshot or something else.**
- Use Google Classroom for your questions. Do not send private messages.

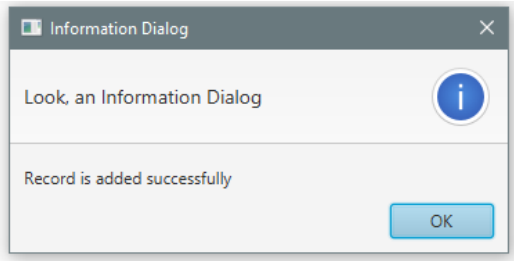
Problem Description:





ADNAN MENDERES UNIVERSITY
CSE 203 Object-Oriented Programming

Alert:



1- Write a java program that stores, retrieves, adds, updates and search addresses as shown in figure. Use a fixed-length string for storing each attribute in the address. Use **random access file(.dat) or text file (.txt)** for reading and writing an address. Assume that the size of id, name, street, city, gender, and zip is 4, 32, 32, 20, 1, 5 characters, respectively.

- Program should perform **add**, **first**, **next**, **previous**, **last**, **updateById**, **SearchById**, **CleanTextFields** operations.
- ID text field should be "read only" and **id** should be set in the code for each record, do not take from user.
- Use **Search/Update ID text Field** as input for **UpdateById** and **SearchById** buttons.
- Create Person class. Data fields are: ID, name, street, city, zip, gender. Create constructor, getters and setters.
- When program starts, read all records from random access file or text file and save each record in Person []. (Person array)
- Update both Person[] and random access file or text file when **add** and **update** operations happen.
- Use Person[] for **first**, **next**, **previous**, **last**, **searchById** operations, do not use random access file.
- Create **Alert** when new record added successfully. You may add some controls (like: alert) in situations like: pressing **"next"** button when viewing last record, pressing **"previous"** button when viewing first record.

2- Submit a report (word) file that explains following questions with a cover page that contains your name and number. **In the report:**

- Describe the problem including major steps, functions for solving the problem and input/output in your own words.
- Draw the UML diagram of your code.
- Describe how you test this program.