

Università degli Studi di Salerno

Dipartimento di Ingegneria dell'Informazione ed Elettrica e
Matematica Applicata



Distributed Programming
Conference

Anno Accademico 2021/2022

Gruppo 9

Renzulli Giuseppe – 0622701514

g.renzulli4@studenti.unisa.it

Vincenzo Salvati – 0622701550

v.salvati10@studenti.unisa.it

Prof De Maio Carmen

Summary

Introduction..... 3

Conference 4

How to use..... 5

Conference

Introduction

This project requests to implement the registration at a conference of the speakers into an organization. This architecture client-server uses the RMI protocol in order to manage or to view sessions about a congress. As far as the program concerned, it provides an interface for each speaker which allow them to enroll in a session, and to view the programs of the various days of the congress, with the interventions of the various sessions.

The Server maintains the programs of the 3 days of the congress on its residence node, each of which is stored in a data structure in which each row corresponds to a session (12 in all for each day). For each session the names of the speakers who have registered are recorded (maximum 5).

Hence, the next paragraphs describe the classes implemented.

```
C:\Users\giuse\.jdk\openjdk-16.0.2\bin\java.exe "-javaagent
ott 21, 2021 10:29:26 AM conference.ServerCongressImpl main
INFO: Creating the remote object
ott 21, 2021 10:29:27 AM conference.ServerCongressImpl main
INFO: rebinding...
ott 21, 2021 10:29:27 AM conference.ServerCongressImpl main
INFO: ...Ready!
Server ready!
|
```

Figure 1 Server CLI

```
C:\Users\giuse\.jdk\openjdk-16.0.2\bin\java.exe "-javaa
ott 21, 2021 10:29:35 AM conference.ClientCongress main
INFO: Creating the remote object
ott 21, 2021 10:29:35 AM conference.ClientCongress main
INFO: Object found!
Chose:
-1.Exit
0.Registration
1.Watch Sessions
```

Figure 2 Client CLI

Conference

The package conference contains all the classes implemented:

- **ClientCongress.java:** firstly, it permits to lockup the object "ServerCongress" from the registry which methos are implemented into a server. So, it allows to have different choices in order to:
 - Registration of the speaker into a session of a specific date;
 - View the session and its registered speakers of a specific date.

This class guide the client in each step as insert a specific date, select the session and other operation like that, but every when you want to come back in order to cancel the current operation it is possible to digit -1.

The program is stopped when you come back from all operation by digit continuously -1 until the CLI shows the message "Close";

- **Program.java:** it is an object which contains the structure for each session distributed into specific date. In particular, it implements two relevant methos:
 - *addSession(Session session):* it allows to add max 12 sessions for a specific date;
 - *addSpeakerForTheSession(int sessionIndex, String speaker):* it allows to add the speaker into a specific session up to five of them.

Both methos return a boolean so that it's possible to know if the operation has been successful;

- **Session.java:** it is an object that take into account the session's name and their speaker who are registered. In particular, it implements an important method called *addSpeaker(String nameSpeaker)* which is used to add a speaker into a session up to five of them;
- **ServerCongress.java:** it is an interface in which has been decelerated the following methods:
 - *getRegisteredDates();*
 - *dateSession(Date date);*
 - *registration(Date date, int session, String speaker);*
 - *program(Date date, boolean speakers);*
- **ServerCongressImpl.java:** it implements the methos of the interface *ServerCongress*:
 - *getRegisteredDates():* returns all recorded date;
 - *dateSession(Date date):* returns if a session is found;
 - *registration(Date date, int session, String speaker):* registers a speaker into a specific session;
 - *program(Date date, Boolean speakers):* print all the sessions of a specific day. In particular there are two sort of modalities by specifying the boolean speakers:
 - if this parameter is true, the method returns only a list of sessions;
 - if this parameter is false the method returns not only a list of session, but also the relative registered speakers.

How to use

To use this application is essential firstly to run the `ServerCongressImpl.java` file by an IDE. The following steps will help you to use the program:

1. **Start the server:** when the server is ready a message "Server ready!" is printed on the IDE's console.

```
C:\Users\giuse\.jdk\openjdk-16.0.2\bin\java.exe "-javaagent:C:\
ott 21, 2021 10:29:26 AM conference.ServerCongressImpl main
INFO: Creating the remote object
ott 21, 2021 10:29:27 AM conference.ServerCongressImpl main
INFO: rebinding...
ott 21, 2021 10:29:27 AM conference.ServerCongressImpl main
INFO: ...Ready!
Server ready!
|
```

2. **Run the client:** it is possible to run the `ClientCongress.java` file by an IDE so that it will appear a CLI which drives you during the use of the application.

```
C:\Users\giuse\.jdk\openjdk-16.0.2\bin\java.exe "-javaa
ott 21, 2021 10:29:35 AM conference.ClientCongress main
INFO: Creating the remote object
ott 21, 2021 10:29:35 AM conference.ClientCongress main
INFO: Object found!
Chose:
-1.Exit
0.Registration
1.Watch Sessions
```

So, it is possible to insert a number in order to choose:

- a) **-1:** to close the application

```
Chose:
-1.Exit
0.Registration
1.Watch Sessions
-1

Exit

Close

Process finished with exit code 0
```

Figure 3 Exit

b) 0: to register a speaker

```
Chose:
-1.Exit
0.Registration
1.Watch Sessions
0

Chose:
-1.Back
0.Insert a date to have a registration
0

Recorded dates:
25/09/2022
22/08/2021

Insert:
- date [gg/mm/yyyy] (or -1 to exit):
25/09/2022

Select a session (or -1 to exit):
0.Congress proof 2
1.Congress proof 3
1

Insert speaker's name for the session (or -1 to exit):
Pippo
Registered correctly.
```

Figure 4 Succeed registration

```
Chose:
-1.Exit
0.Registration
1.Watch Sessions
0

Chose:
-1.Back
0.Insert a date to have a registration
0

Recorded dates:
22/08/2021
25/09/2022

Insert:
- date [gg/mm/yyyy] (or -1 to exit):
22/08/2021

Select a session (or -1 to exit):
0.Congress proof 1
0

Insert speaker's name for the session (or -1 to exit):
Pluto
Registration not carried out: session full.
```

Figure 5 Failed registration

- c) 1: to watch the sessions for a day

```
Chose:
-1.Exit
0.Registration
1.Watch Sessions
1

Chose:
-1.Back
0.Insert a date to analyze specific session
0

Recorded dates:
22/08/2021
25/09/2022

Insert:
- date [gg/mm/yyyy] (or -1 to exit)):
25/09/22

--- Sessions ---

Name Session: Congress proof 2
1° speaker: Pippo
2° speaker: Pluto
3° speaker: Paperino

Name Session: Congress proof 3
1° speaker: Pippo
2° speaker: Pluto
3° speaker: Paperino
4° speaker: Topolino
5° speaker: Pippo
```

Figure 6 Printed session

P.S.

- There are two established congresses 'date':
 - 22/08/2021: which contains only one full session;
 - 25/09/2022: which contains two sessions respectively with three and four registered speakers;
- Whenever you need to choose a recorded date for any operation that require that, it is printed a list of them for having a view of them;
- It is possible to back whenever you want digiting -1;
- More speakers can have the same name in the same session.