

Service-oriented Architecture INT 3505 1

Homework 2: RMI programming

1. Team member

	Student name	ID
1	Nguyễn Việt Minh Nghĩa	15021358

2. How to run

```
# Start rmiregistry daemon  
rmiregistry&
```

```
# Start server  
javac rmi_ex/Server.java  
java rmi_ex.Server
```

```
# Start client  
javac rmi_ex/Client.java  
java rmi_ex.Client
```

Enter the ID to get the information.

3. Screenshot

```

1 package rml_ex;
2
3 import java.rmi.Remote;
4 import java.rmi.RemoteException;
5
6
7 interface IGetStudent extends Remote {
8     public Student getStudentFromID(String id) throws RemoteException;
9 }
10

```

```

1 package rml_ex;
2
3 import java.io.Serializable;
4
5
6 public class Student implements Serializable {
7     public String name;
8     public String id;
9     public String className;
10    public String phone;
11    public String email;
12    public String address;
13
14
15    public Student(String name, String id, String className,
16                    String phone, String email, String address) {
17        this.name = name;
18        this.id = id;
19        this.className = className;
20        this.phone = phone;
21        this.email = email;
22        this.address = address;
23    }
24
25    @Override
26    public String toString() {
27        StringBuilder b = new StringBuilder();
28
29        b.append("ID: ").append(this.id).append("\n");
30        b.append("Name: ").append(this.name).append("\n");
31        b.append("Class: ").append(this.className).append("\n");
32        b.append("Phone: ").append(this.phone).append("\n");
33    }
34
35    @Override
36    public static void main(String[] args) {
37        // First int PORT = 0; // Anonymous port? Not but 1099?
38        Server server = new Server();
39        IGetStudent stub = (IGetStudent) UnicastRemoteObject
40            .exportObject(server, PORT); // Alternatively, extends with Un
41        Registry registry = LocateRegistry.getRegistry(); // and omit this exporting line
42        registry.bind("server", stub);
43        } catch (Exception e) {
44            e.printStackTrace();
45        }
46    }
47
48

```

```

1 package rml_ex;
2
3 import java.io.Serializable;
4
5
6 public class Student implements Serializable {
7     public String name;
8     public String id;
9     public String className;
10    public String phone;
11    public String email;
12    public String address;
13
14
15    public Student(String name, String id, String className,
16                    String phone, String email, String address) {
17        this.name = name;
18        this.id = id;
19        this.className = className;
20        this.phone = phone;
21        this.email = email;
22        this.address = address;
23    }
24
25    @Override
26    public String toString() {
27        StringBuilder b = new StringBuilder();
28
29        b.append("ID: ").append(this.id).append("\n");
30        b.append("Name: ").append(this.name).append("\n");
31        b.append("Class: ").append(this.className).append("\n");
32        b.append("Phone: ").append(this.phone).append("\n");
33    }
34
35    @Override
36    public static void main(String[] args) {
37        // First int PORT = 0; // Anonymous port? Not but 1099?
38        Server server = new Server();
39        IGetStudent stub = (IGetStudent) UnicastRemoteObject
40            .exportObject(server, PORT); // Alternatively, extends with Un
41        Registry registry = LocateRegistry.getRegistry(); // and omit this exporting line
42        registry.bind("server", stub);
43        } catch (Exception e) {
44            e.printStackTrace();
45        }
46    }
47
48

```

```

1 package rml_ex;
2
3 import java.io.Serializable;
4
5
6 public class Student implements Serializable {
7     public String name;
8     public String id;
9     public String className;
10    public String phone;
11    public String email;
12    public String address;
13
14
15    public Student(String name, String id, String className,
16                    String phone, String email, String address) {
17        this.name = name;
18        this.id = id;
19        this.className = className;
20        this.phone = phone;
21        this.email = email;
22        this.address = address;
23    }
24
25    @Override
26    public String toString() {
27        StringBuilder b = new StringBuilder();
28
29        b.append("ID: ").append(this.id).append("\n");
30        b.append("Name: ").append(this.name).append("\n");
31        b.append("Class: ").append(this.className).append("\n");
32        b.append("Phone: ").append(this.phone).append("\n");
33    }
34
35    @Override
36    public static void main(String[] args) {
37        // First int PORT = 0; // Anonymous port? Not but 1099?
38        Server server = new Server();
39        IGetStudent stub = (IGetStudent) UnicastRemoteObject
40            .exportObject(server, PORT); // Alternatively, extends with Un
41        Registry registry = LocateRegistry.getRegistry(); // and omit this exporting line
42        registry.bind("server", stub);
43        } catch (Exception e) {
44            e.printStackTrace();
45        }
46    }
47
48

```

```

1 package rml_ex;
2
3 import java.io.Serializable;
4
5
6 public class Student implements Serializable {
7     public String name;
8     public String id;
9     public String className;
10    public String phone;
11    public String email;
12    public String address;
13
14
15    public Student(String name, String id, String className,
16                    String phone, String email, String address) {
17        this.name = name;
18        this.id = id;
19        this.className = className;
20        this.phone = phone;
21        this.email = email;
22        this.address = address;
23    }
24
25    @Override
26    public String toString() {
27        StringBuilder b = new StringBuilder();
28
29        b.append("ID: ").append(this.id).append("\n");
30        b.append("Name: ").append(this.name).append("\n");
31        b.append("Class: ").append(this.className).append("\n");
32        b.append("Phone: ").append(this.phone).append("\n");
33    }
34
35    @Override
36    public static void main(String[] args) {
37        // First int PORT = 0; // Anonymous port? Not but 1099?
38        Server server = new Server();
39        IGetStudent stub = (IGetStudent) UnicastRemoteObject
40            .exportObject(server, PORT); // Alternatively, extends with Un
41        Registry registry = LocateRegistry.getRegistry(); // and omit this exporting line
42        registry.bind("server", stub);
43        } catch (Exception e) {
44            e.printStackTrace();
45        }
46    }
47
48

```

```

1 package rml_ex;
2
3 import java.io.Serializable;
4
5
6 public class Student implements Serializable {
7     public String name;
8     public String id;
9     public String className;
10    public String phone;
11    public String email;
12    public String address;
13
14
15    public Student(String name, String id, String className,
16                    String phone, String email, String address) {
17        this.name = name;
18        this.id = id;
19        this.className = className;
20        this.phone = phone;
21        this.email = email;
22        this.address = address;
23    }
24
25    @Override
26    public String toString() {
27        StringBuilder b = new StringBuilder();
28
29        b.append("ID: ").append(this.id).append("\n");
30        b.append("Name: ").append(this.name).append("\n");
31        b.append("Class: ").append(this.className).append("\n");
32        b.append("Phone: ").append(this.phone).append("\n");
33    }
34
35    @Override
36    public static void main(String[] args) {
37        // First int PORT = 0; // Anonymous port? Not but 1099?
38        Server server = new Server();
39        IGetStudent stub = (IGetStudent) UnicastRemoteObject
40            .exportObject(server, PORT); // Alternatively, extends with Un
41        Registry registry = LocateRegistry.getRegistry(); // and omit this exporting line
42        registry.bind("server", stub);
43        } catch (Exception e) {
44            e.printStackTrace();
45        }
46    }
47
48

```

```

1 package rml_ex;
2
3 import java.io.Serializable;
4
5
6 public class Student implements Serializable {
7     public String name;
8     public String id;
9     public String className;
10    public String phone;
11    public String email;
12    public String address;
13
14
15    public Student(String name, String id, String className,
16                    String phone, String email, String address) {
17        this.name = name;
18        this.id = id;
19        this.className = className;
20        this.phone = phone;
21        this.email = email;
22        this.address = address;
23    }
24
25    @Override
26    public String toString() {
27        StringBuilder b = new StringBuilder();
28
29        b.append("ID: ").append(this.id).append("\n");
30        b.append("Name: ").append(this.name).append("\n");
31        b.append("Class: ").append(this.className).append("\n");
32        b.append("Phone: ").append(this.phone).append("\n");
33    }
34
35    @Override
36    public static void main(String[] args) {
37        // First int PORT = 0; // Anonymous port? Not but 1099?
38        Server server = new Server();
39        IGetStudent stub = (IGetStudent) UnicastRemoteObject
40            .exportObject(server, PORT); // Alternatively, extends with Un
41        Registry registry = LocateRegistry.getRegistry(); // and omit this exporting line
42        registry.bind("server", stub);
43        } catch (Exception e) {
44            e.printStackTrace();
45        }
46    }
47
48

```