**Лекция 7**

Следующий пример будет отличаться от предыдущего тем, что в нем будут реализованы методы, оперирующие объектными ссылками. В отличие от предыдущего примера, определим Card как интерфейс, обладающий методами изменения баланса. Объекты типа Card будут создаваться в процессе работы системы, и ссылки на них будут возвращаться клиенту соответствующими методами.

Таким образом, доступ к объектам этого типа будет осуществляться не посредством обращения к сервису имен, а посредством получения объектной ссылки.

**Файл BillingService.idl**

Файл **BillingService.idl** содержит описание двух интерфейсов: **Card** и **BillingService**.Интерфейс **Card** содержит три метода: **addMoney** (добавляет денежные средства на карту), **subMoney** (снимает денежные средства с карты), **getCardBalance** (показывает текущий баланс карты). Интерфейс **BillingService** включает два метода: **addNewCard** (добавляет новую карту) и **getCard** (возвращает объектную ссылку на карту с указанным номером).

В [примере 8.1](https://www.intuit.ru/studies/courses/633/489/lecture/11084?page=1#example.8.1) приведено IDL -описание сервера **BillingService**.

1 // BillingService.idl

2 // IDL-описание BillingService

3 module BillingServiceModule

4 {

5 interface Card

6 {

7 void addMoney(in double money);

8 void subMoney(in double money);

9 double getBalance();

10 };

11 interface BillingService

12 {

13 Card getCard(in string card);

14 Card addNewCard(in string personName, in string card);

15 };

16 };

Листинг 8.1. IDL-описание сервера BillingService

Следующим шагом в реализации нашего приложения будет генерация из IDL -определения вспомогательных классов.

После компиляции файла **BillingService.idl** компилятор Java-IDL создает следующие файлы на стороне сервера в вновь созданной директории **BillingServiceModule**:

BillingService.java,

BillingServiceHelper.java,

BillingServiceHolder.java,

BillingServiceOperations.java,

BillingServicePOA.java,

\_BillingServiceStub.java,

Card.java,

CardHelper.java,

CardHolder.java,

CardOperations.java,

CardPOA.java,

\_CardStub.java.

Рассмотрим подробнее эти файлы (в этих файлах определены классы и интерфейсы языка java ) и то, как мы можем их использовать при создании нашего приложения.

**BillingService.java** и **BillingServiceOperations.java** - интерфейсы. **BillingService.java** ([пример 8.2](https://www.intuit.ru/studies/courses/633/489/lecture/11084?page=1#example.8.2)) представляет собой интерфейс **BillingService**.

1 package com.asw.corba.ex2.BillingServiceModule;

2

3

4 /\*\*

5 \* com/asw/corba/ex2/BillingServiceModule/BillingService.java .

6 \* Generated by the IDL-to-Java compiler (portable), version "3.1"

7 \* from com/asw/corba/ex2/BillingService.idl

8 \* 16 Июль 2006 г. 15:01:11 MSD

9 \*/

10

11 public interface BillingService extends BillingServiceOperations,

12 org.omg.CORBA.Object, org.omg.CORBA.portable.IDLEntity

13 {

14 } // interface BillingService

Листинг 8.2. Интерфейс BillingService, созданный idlj

В строках 11 и 12 объявлены интерфейс **BillingService** и три интерфейса, от которых наследует **BillingService**.Как уже говорилось, два из этих трех интерфейсов являются определенными в CORBA типами, от которых должны наследовать все CORBA -совместимые объекты, **org.omg.CORBA.Object** и **org.omg.CORBA.portable.IDLEntity**.Третий интерфейс - **BillingServiceOperations** ([пример 8.3](https://www.intuit.ru/studies/courses/633/489/lecture/11084?page=1#example.8.3)) - создается на основе IDL -описания и объявляет открытые операции данного сервера. **BillingService** определяет базовый класс, который наследует от **BillingServiceOperations** и от CORBA -интерфейсов, упомянутых ранее. **BillingServiceOperations** объявляет два метода, изначально определенные в IDL.

1 package com.asw.corba.ex2.BillingServiceModule;

2

3

4 /\*\*

5 \* com/asw/corba/ex2/BillingServiceModule/BillingServiceOperations.java .

6 \* Generated by the IDL-to-Java compiler (portable), version "3.1"

7 \* from com/asw/corba/ex2/BillingService.idl

8 \* 16 Июль 2006 г. 15:01:11 MSD

9 \*/

10

11 // определение CORBA-совместимого сервиса

12 public interface BillingServiceOperations

13 {

14 com.asw.corba.ex2.BillingServiceModule.Card getCard (String card);

15 com.asw.corba.ex2.BillingServiceModule.Card addNewCard

(String personName, String card);

16 } // interface BillingServiceOperations

Листинг 8.3. Интерфейс BillingServiceOperations, созданный idlj

Файлы **BillingServiceHelper.java** ([пример 8.4](https://www.intuit.ru/studies/courses/633/489/lecture/11084?page=1#example.8.4)) и **BillingServiceHolder.java** ([пример 8.5](https://www.intuit.ru/studies/courses/633/489/lecture/11084?page=1#example.8.5)) содержат методы приведения типов и методы маршаллинга/демаршаллинга сообщений.

1 package com.asw.corba.ex2.BillingServiceModule;

2

3

4 /\*\*

5 \* com/asw/corba/ex2/BillingServiceModule/BillingServiceHelper.java

6 \* Generated by the IDL-to-Java compiler (portable), version "3.1"

7 \* from com/asw/corba/ex2/BillingService.idl

8 \* 16 Июля 2006 г. 15:01:11 MSD

9 \*/

10

11 // определение класса BillingServiceHelper

12 abstract public class BillingServiceHelper

13 {

14 private static String \_id =

"IDL:BillingServiceModule/BillingService:1.0";

15

16 public static void insert (org.omg.CORBA.Any a,

com.asw.corba.ex2.BillingServiceModule.BillingService that)

17 {

18 org.omg.CORBA.portable.OutputStream out = a.create\_output\_stream ();

19 a.type (type ());

20 write (out, that);

21 a.read\_value (out.create\_input\_stream (), type ());

22 }

23

24 public static com.asw.corba.ex2.BillingServiceModule.BillingService

extract (org.omg.CORBA.Any a)

25 {

26 return read (a.create\_input\_stream ());

27 }

28

29 private static org.omg.CORBA.TypeCode\_\_typeCode = null;

30

31 synchronized public static org.omg.CORBA.TypeCode type ()

32 {

33 if (\_\_typeCode == null)

34 {

35 \_\_typeCode = org.omg.CORBA.ORB.init ().create\_interface\_tc

(com.asw.corba.ex2.BillingServiceModule.BillingServiceHelper.id (),

"BillingService");

36 }

37 return \_\_typeCode;

38 }

39

40 public static String id ()

41 {

42 return \_id;

43 }

44

45 public static com.asw.corba.ex2.BillingServiceModule.BillingService

read (org.omg.CORBA.portable.InputStream istream)

46 {

47 return narrow (istream.read\_Object (\_BillingServiceStub.class));

48 }

49

50 public static void write (org.omg.CORBA.portable.OutputStream ostream,

com.asw.corba.ex2.BillingServiceModule.BillingService value)

51 {

52 ostream.write\_Object ((org.omg.CORBA.Object) value);

53 }

54

55 public static com.asw.corba.ex2.BillingServiceModule.BillingService

narrow (org.omg.CORBA.Object obj)

56 {

57 if (obj == null)

58 return null;

59 else if (obj instanceof

com.asw.corba.ex2.BillingServiceModule.BillingService)

60 return (com.asw.corba.ex2.BillingServiceModule.BillingService)obj;

61 else if (!obj.\_is\_a (id ()))

62 throw new org.omg.CORBA.BAD\_PARAM ();

63 else

64 {

65 org.omg.CORBA.portable.Delegate delegate =

((org.omg.CORBA.portable.ObjectImpl)obj).\_get\_delegate ();

66 com.asw.corba.ex2.BillingServiceModule.\_BillingServiceStub stub =

new com.asw.corba.ex2.BillingServiceModule.\_BillingServiceStub ();

67 stub.\_set\_delegate(delegate);

68 return stub;

69 }

70 }

71 }

Листинг 8.4. Класс BillingServiceHelper, созданный idlj

1 package com.asw.corba.ex2.BillingServiceModule;

2

3 /\*\*

4 \* com/asw/corba/ex2/BillingServiceModule/BillingServiceHolder.java .

5 \* Generated by the IDL-to-Java compiler (portable), version "3.1"

6 \* from com/asw/corba/ex2/BillingService.idl

7 \* 16 Июль 2006 г. 15:01:11 MSD

8 \*/

9

10 // определение класса BillingServiceHolder

11 public final class BillingServiceHolder implements

org.omg.CORBA.portable.Streamable

12 {

13 public com.asw.corba.ex2.BillingServiceModule.BillingService value

= null;

14

15 public BillingServiceHolder ()

16 {

17 }

18

19 public BillingServiceHolder

(com.asw.corba.ex2.BillingServiceModule.BillingService initialValue)

20 {

21 value = initialValue;

22 }

23

24 public void \_read (org.omg.CORBA.portable.InputStream i)

25 {

26 value =

com.asw.corba.ex2.BillingServiceModule.BillingServiceHelper.read (i);

27 }

28

29 public void \_write (org.omg.CORBA.portable.OutputStream o)

30 {

31 com.asw.corba.ex2.BillingServiceModule.BillingServiceHelper.write

(o, value);

32 }

33

34 public org.omg.CORBA.TypeCode \_type ()

35 {

36 return

com.asw.corba.ex2.BillingServiceModule.BillingServiceHelper.type ();

37 }

38 }

Листинг 8.5. Класс BillingServiceHolder, созданный idlj

Файл **\_BillingServiceStub.java** ([пример 8.6](https://www.intuit.ru/studies/courses/633/489/lecture/11084?page=2#example.8.6)) содержит клиентскую заглушку. Класс **BillingServiceStub** наследует от **org.omg.CORBA.portable. ObjectImpl** и реализует интерфейс **BillingService**.При вызове клиентом методов этого класса происходит упаковка вызова (и параметров) и их передача по сети серверному классу.

1 package com.asw.corba.ex2.BillingServiceModule;

2

3 /\*\*

4 \* com/asw/corba/ex2/BillingServiceModule/\_BillingServiceStub.java.

5 \* Generated by the IDL-to-Java compiler (portable), version "3.1"

6 \* from com/asw/corba/ex2/BillingService.idl

7 \* 16 Июль 2006 г. 15:01:11 MSD

8 \*/

9

10 // определение класса \_BillingServiceStub

11 public class \_BillingServiceStub extends org.omg.CORBA.

portable.ObjectImpl implements

com.asw.corba.ex2.BillingServiceModule.BillingService

12 {

13 public com.asw.corba.ex2.BillingServiceModule.Card getCard (String card)

14 {

15 org.omg.CORBA.portable.InputStream $in = null;

16 try {

17 org.omg.CORBA.portable.OutputStream $out =

\_request ("getCard", true);

18 $out.write\_string (card);

19 $in = \_invoke ($out);

20 com.asw.corba.ex2.BillingServiceModule.Card $result =

com.asw.corba.ex2.BillingServiceModule.CardHelper.read ($in);

21 return $result;

22 } catch (org.omg.CORBA.portable.ApplicationException $ex) {

23 $in = $ex.getInputStream ();

24 String \_id = $ex.getId ();

25 throw new org.omg.CORBA.MARSHAL (\_id);

26 } catch (org.omg.CORBA.portable.RemarshalException $rm) {

27 return getCard (card );

28 } finally {

29 \_releaseReply ($in);

30 }

31 } // getCard

32

33 public com.asw.corba.ex2.BillingServiceModule.Card addNewCard

(String personName, String card)

34 {

35 org.omg.CORBA.portable.InputStream $in = null;

36 try {

37 org.omg.CORBA.portable.OutputStream $out =

\_request ("addNewCard", true);

38 $out.write\_string (personName);

39 $out.write\_string (card);

40 $in = \_invoke ($out);

41 com.asw.corba.ex2.BillingServiceModule.Card $result =

com.asw.corba.ex2.BillingServiceModule.CardHelper.read ($in);

42 return $result;

43 } catch (org.omg.CORBA.portable.ApplicationException $ex) {

44 $in = $ex.getInputStream ();

45 String \_id = $ex.getId ();

46 throw new org.omg.CORBA.MARSHAL (\_id);

47 } catch (org.omg.CORBA.portable.RemarshalException $rm) {

48 return addNewCard (personName, card );

49 } finally {

50 \_releaseReply ($in);

51 }

52 } // addNewCard

53

54 // Type-specific CORBA::Object operations

55 private static String[]\_\_ids = {

56 "IDL:BillingServiceModule/BillingService:1.0"};

57 public String[] \_ids ()

58 {

59 return (String[])\_\_ids.clone ();

60 }

61

62 private void readObject (java.io.ObjectInputStream s)

throws java.io.IOException

63 {

64 String str = s.readUTF ();

65 String[] args = null;

66 java.util.Properties props = null;

67 org.omg.CORBA.Object obj = org.omg.CORBA.ORB.init

(args, props).string\_to\_object (str);

68 org.omg.CORBA.portable.Delegate delegate =

((org.omg.CORBA.portable.ObjectImpl) obj).\_get\_delegate ();

69 \_set\_delegate (delegate);

70 }

71

72 private void writeObject (java.io.ObjectOutputStream s)

throws java.io.IOException

73 {

74 String[] args = null;

75 java.util.Properties props = null;

76 String str = org.omg.CORBA.ORB.init (args, props).object\_to\_string

(this);

77 s.writeUTF (str);

78 }

79 } // class \_BillingServiceStub

Листинг 8.6. Класс BillingServiceStub, созданный idlj

И последний файл **BillingServicePOA.java** ([пример 8.7](https://www.intuit.ru/studies/courses/633/489/lecture/11084?page=2#example.8.7)) содержит класс, выполняющий всю техническую работу по диспетчеризации сообщений клиента, - этот класс будет являться предком для нашего серверного класса, в котором мы переопределим методы, выполняющие содержательные операции. Класс **BillingServicePOA** наследует от **org.omg.PortableServer.Servant** и реализует CORBA -интерфейс **(InvokeHandler)** и интерфейс **BillingServiceOperations**.Интерфейс **InvokeHandler** объявляет метод invoke, который является реализацией **BillingServicePOA**.

1 package com.asw.corba.ex2.BillingServiceModule;

2

3 /\*\*

4 \* com/asw/corba/ex2/BillingServiceModule/BillingServicePOA.java .

5 \* Generated by the IDL-to-Java compiler (portable), version "3.1"

6 \* from com/asw/corba/ex2/BillingService.idl

7 \* 16 Июль 2006 г. 15:01:11 MSD

8 \*/

9

10 // определение класса BillingServicePOA

11 public abstract class BillingServicePOA extends

org.omg.PortableServer.Servant

12 implements

com.asw.corba.ex2.BillingServiceModule.BillingServiceOperations,

org.omg.CORBA.portable.InvokeHandler

13 {

14 // Constructors

15 private static java.util.Hashtable \_methods = new java.util.Hashtable ();

16 static

17 {

18 \_methods.put ("getCard", new java.lang.Integer (0));

19 \_methods.put ("addNewCard", new java.lang.Integer (1));

20 }

21

22 public org.omg.CORBA.portable.OutputStream \_invoke (String $method,

23 org.omg.CORBA.portable.InputStream in,

24 org.omg.CORBA.portable.ResponseHandler $rh)

25 {

26 org.omg.CORBA.portable.OutputStream out = null;

27 java.lang.Integer \_\_method = (java.lang.Integer)\_methods.get ($method);

28 if (\_\_method == null)

29 throw new org.omg.CORBA.BAD\_OPERATION (0,

org.omg.CORBA.CompletionStatus.COMPLETED\_MAYBE);

30

31 switch \_\_method.intValue ())

32 {

33 case 0: // BillingServiceModule/BillingService/getCard

34 {

35 String card = in.read\_string ();

36 com.asw.corba.ex2.BillingServiceModule.Card $result = null;

37 $result = this.getCard (card);

38 out = $rh.createReply();

39 com.asw.corba.ex2.BillingServiceModule.CardHelper.write (out, $result);

40 break;

41 }

42

43 case 1: // BillingServiceModule/BillingService/addNewCard

44 {

45 String personName = in.read\_string ();

46 String card = in.read\_string ();

47 com.asw.corba.ex2.BillingServiceModule.Card $result = null;

48 $result = this.addNewCard (personName, card);

49 out = $rh.createReply();

50 com.asw.corba.ex2.BillingServiceModule.CardHelper.write (out, $result);

51 break;

52 }

53

54 default:

55 throw new org.omg.CORBA.BAD\_OPERATION

(0, org.omg.CORBA.CompletionStatus.COMPLETED\_MAYBE);

56 }

57 return out;

58 } // \_invoke

59

60 // Type-specific CORBA::Object operations

61 private static String[] \_\_ids =

{"IDL:BillingServiceModule/BillingService:1.0"};

62

63 public String[] \_all\_interfaces

(org.omg.PortableServer.POA poa, byte[] objectId)

64 {

65 return (String[]) \_\_ids.clone ();

66 }

67

68 public BillingService \_this()

69 {

70 return BillingServiceHelper.narrow(super.\_this\_object());

71 }

72

73 public BillingService \_this(org.omg.CORBA.ORB orb)

74 {

75 return BillingServiceHelper.narrow(super.\_this\_object(orb));

76 }

77

78 } // class BillingServicePOA

Листинг 8.7. Класс BillingServicePOA, созданный idlj