Урок 119. ПР Сериализация

1 ноября 2017 г.

13:03

Практическая работа: "Сериализация и десериализация".

План работы.

1. Создать объект класса **Date** или **GregorianCalendar** с текущей датой и временем.

Вывод на экран:

вариант : 
2 вариант: 
З вариант: 
4 вариант: 
S вадант: 
б вадант: 
Лед Ncv 
01 
ЕЕТ 2017 
Ncverber 2017 
ноябрь 2017 
ноября 2017 
Листопада 2017 
01 умстспада 2017 
(О. S бала) 
(2 бала) 

1. Cериализовать объект в файл **date.ser** (дополнительные баллы за использование try-catch-finally)

1. Через 5 секунд десериализовать и сравнить с текущим временем.

1. Создать класс по образцу, согласно варианта:

String name; 
int weight; 
String breed; 
public 
Dog (String name, int weight, 
String breed) 
this. weigh t 
this. breed 
weight ; 
= breed; 

S tuden t 
String surname; 
String subject; 
int mark; 
boolean isPassed; 
public 
Student (String surname , 
this. su rname 
— surname; 
this. subject = subject; 
this. mark = mark; 
this. isPassed = is Passed; 
String subject , 
int mark, 
boolean is Passed) 

Car 
int price; 
String model; 
double power ; 
public 
Car (int price, String model, 
double power) 
this. price 
this. model 
this. power 
price; 
— model; 
power ; 

1. Привести описание класса к требованиям JavaBeanNaming

1. Сериализовать 2 екземпляра этого класса и десериализовать их, используя тот же файл **date.ser**

1. Вывести на экран названия исключений, которые возникают при сериализации и десериализации.

Джерело: <<https://schoolboyprog10.blogspot.com/p/blog-page_82.html>>

Решение:

**import**java.io.\*;

**import**java.text.\*;

**import**java.util.\*;

**public class** Dispatcher{

**public static void** main(String[]args) **throws** IOException, ClassNotFoundException{

**int** seconds;

**int** minutes;

**int** hours;

**int** day;

String month;

**int** year;

*//\*\*\*\*\*\*\*\*\*1\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

System.***out***.print(**"1вариант:"**);

Date d = **new** Date();

System.***out***.println(d.toString()+**"(0.5бала)"**);

*//\*\*\*\*\*\*\*\*\*2\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

GregorianCalendar gc = **new** GregorianCalendar(); *//createinstanceofGrCalendar*

System.***out***.print(**"2вариант:"**);

seconds=gc.get(Calendar.***SECOND***);

minutes=gc.get(Calendar.***MINUTE***);

hours=gc.get(Calendar.***HOUR***);

day=gc.get(Calendar.***DAY\_OF\_MONTH***);

month=gc.getDisplayName(Calendar.***MONTH***,Calendar.***LONG***,Locale.*getDefault*());

year=gc.get(Calendar.***YEAR***);

System.***out***.println(hours+**":"**+minutes+**":"**+seconds+**","**+day+**""**+month+**""**+year);

*//\*\*\*\*\*\*\*\*\*3\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

System.***out***.print(**"3 вариант:"**);

month=**new** SimpleDateFormat(**"MMMM"**).format(gc.getTime());

System.***out***.println(hours+**":"**+minutes+**":"**+seconds+**","**+day+**""**+month+**""**+year);

*//\*\*\*\*\*\*\*\*\*4\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

System.***out***.print(**"4 вариант:"**);

String[]months=**new** DateFormatSymbols().getMonths();

month=months[gc.get(Calendar.***MONTH***)];

System.***out***.println(hours+**":"**+minutes+**":"**+seconds+**","**+day+**""**+month+**""**+year);

*//\*\*\*\*\*\*\*\*\*5\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

System.***out***.print(**"5варіант:"**);

month=*getMonthName*(gc.get(Calendar.***MONTH***));

System.***out***.println(hours+**":"**+minutes+**":"**+seconds+**","**+day+**""**+month+**""**+year);

*//\*\*\*\*\*\*\*\*\*6\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

System.***out***.print(**"6варіант:"**);

Localeb = **new** Locale(**"uk"**,**"UA"**);

SimpleDateFormat sdf=**new** SimpleDateFormat(**"HH:mm:ss,ddMMMMyyyy"**,b);

System.***out***.println(sdf.format(gc.getTime())+**"(2бала)"**);

System.***out***.println();

*//createstreamwithoutputtowriteinstance*

ObjectOutputStream out = **new** ObjectOutputStream(**new** FileOutputStream(**"objects.dat"**));

out.writeObject(gc);

out.close();

System.***out***.println(**"Ihavewritten:"**);

System.***out***.println(**"AGrCalendarobject:"**+gc);

Thread.*sleep*(2000); // задержка времени

*//wait5seconds*

**Int** k=1;

**Int** secondsTemp = seconds;

**do**{

**if**(secondsTemp>54){

secondsTemp = 60 - secondsTemp;

k=-1;

}

*//System.out.println(seconds+""+secondsTemp);*

}**while**(**new** GregorianCalendar().get(Calendar.***SECOND***)-secondsTemp\*k!=5);

*//createstreamwithinputtoreadinstance*

ObjectInputStream in = **new** ObjectInputStream(**new** FileInputStream( **"objects.dat"**));

GregorianCalendar gcLoad=(GregorianCalendar)in.readObject();

System.***out***.println(**"I have read:"**);

System.***out***.println(**"A GrCalendar object:"**+gcLoad);

in.close();

*//equaltwoGrCalendars*

System.***out***.println(gcLoad.get(Calendar.***SECOND***)+**""**+**new**GregorianCalendar().get(Calendar.***SECOND***));

*//create instanceof class Random*

Random r=**new** Random();

r.setNumberToRandom();

*//createinstancesofclassDog*

Dogd1 = **new** Dog(**"Jack"**,10,**"Poodle"**);

Dogd2 = **new** Dog(**"Rex"**,8,**"Shepherd"**);

*//createstreamwithoutputtowriteinstanceDog*

ObjectOutputStreamout2=**new**ObjectOutputStream(**new**FileOutputStream(**"objects.dat"**));

out2.writeObject(d1); //запись

out2.writeObject(d2);

out2.close();

System.***out***.println(**"I have written:"**);

System.***out***.println(**"A Dog object:"**+d1.**name**);

System.***out***.println(**"A Dog object:"**+d2.**name**);

*//createstreamwithinputtoreadinstanceDog*

ObjectInputStreamin2=**null**;

**try**{

in2=**new**ObjectInputStream(**new**FileInputStream(**"objects.dat"**));

Dogd1Load1=(Dog)in2.readObject(); //чтение

Dogd1Load2=(Dog)in2.readObject();

System.***out***.println(**"I have read:"**);

System.***out***.println(**"A Dog object:"**+d1Load1.**name**);

System.***out***.println(**"A Dog object:"**+d1Load2.**name**);

}**catch**(ClassCastExceptione){

System.***err***.println(**"ErrorcastingobjecttoaRectangle"**);

}**catch**(ClassNotFoundExceptione){

System.***err***.println(**"Classnotfound"**);

}**catch**(IOExceptione){

System.***err***.println(**"Errorreadingobject:"**+e.getMessage());

}**finally**{

**try**{

in.close();

}**catch**(IOExceptione){

System.***err***.println(e.getMessage());

}

}

}

*//methodfor4variantofGrCalendar*

**publicstatic**StringgetMonthName(**int**month){

String[]monthNames={**"Січня"**,**"Лютого"**,**"Березня"**,**"Квітня"**,**"Травня"**,**"Червня"**,**"Липня"**,**"Серпня"**,**"Вересня"**,**"Жовтня"**,**"Листопада"**,**"Грудня"**};

**return**monthNames[month];

}

}

**class**Random**implements**Serializable{

**privateintnumber**;

**publicvoid**setNumberToRandom(){

**this**.**number**=(**int**)Math.*random*()\*100;

}

}

**class**Dog**implements**Serializable{

String**name**;

**intweight**;

String**breed**;

**public**Dog(Stringname,**int**weight,Stringbreed){

**this**.**name**=name;

**this**.**weight**=weight;

**this**.**breed**=breed;

}

}

**class**Student{

String**surname**;

String**subject**;

**intmark**;

**booleanisPassed**;

**public**Student(Stringsurname,Stringsubject,**int**mark,**boolean**isPassed){

**this**.**surname**=surname;

**this**.**subject**=subject;

**this**.**mark**=mark;

**this**.**isPassed**=isPassed;

}

}

**class**Car{

**intprice**;

String**model**;

**doublepower**;

**public**Car(**int**price,Stringmodel,**double**power){

**this**.**price**=price;

**this**.**model**=model;

**this**.**power**=power;

}

}