

11. Lekcja

Operacje na wrapperach i datach

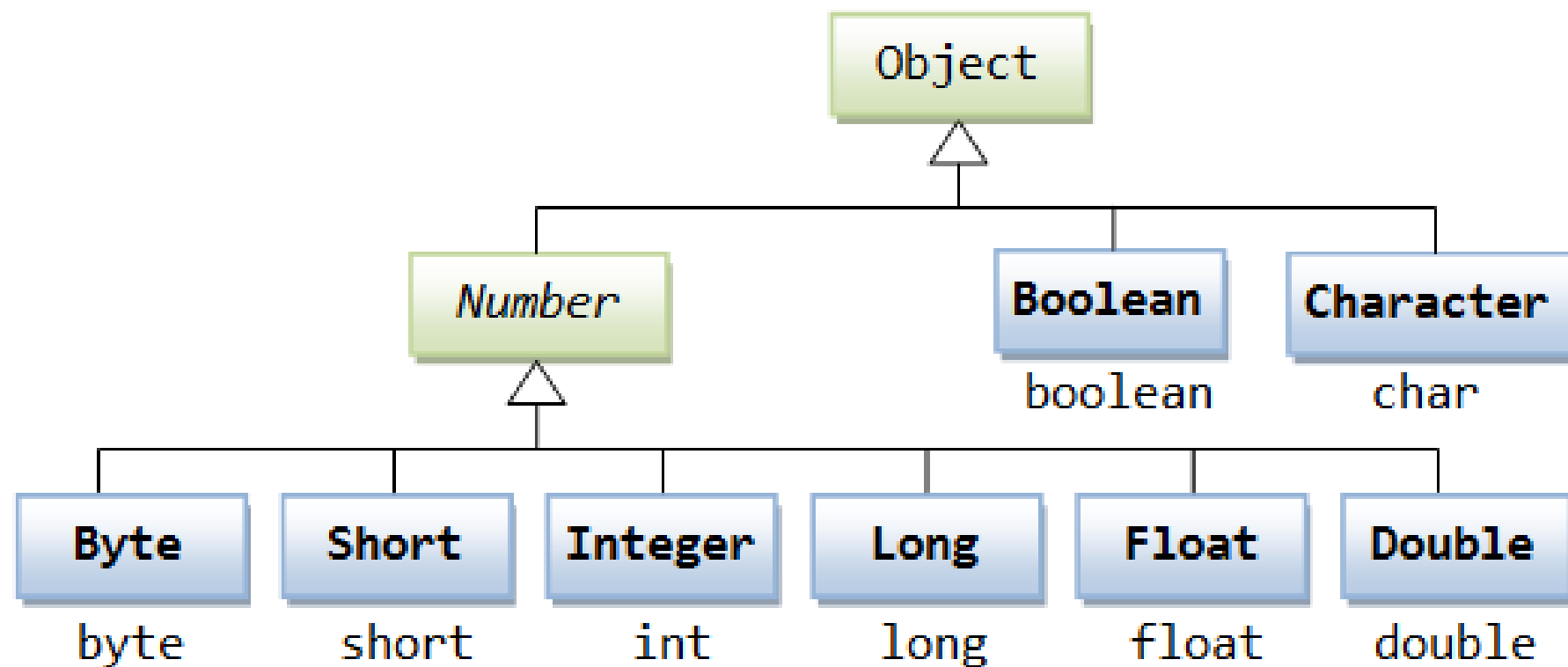


Wrappery



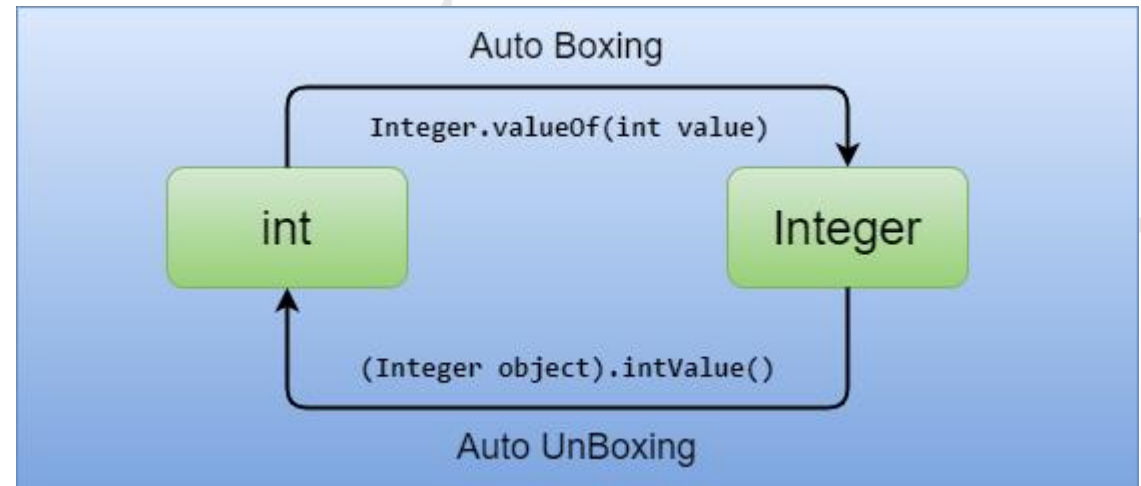
- To specjalne klasy służące do przechowywania typów prostych.
- Dzięki wrapperą jesteśmy w stanie traktować typy proste jak obiekty.

Hierarchia Wrapperów



Pakowanie i rozpakowywanie

- Każda z klas wrapperów posiada statyczną metodę „valueOf” służącą do zapakowania typu prostego.
- Rozpakować taki obiekt możemy do różnych innych typów prostych.



//Boxing

```
Integer integerFromInt = Integer.valueOf(1);  
Integer integerFromString = Integer.valueOf("2");
```

//Unboxing

```
byte byteValueFromInt = integerFromInt.byteValue();  
int intValueFromInt = integerFromInt.intValue();  
String stringValueFromInt = integerFromInt.toString();  
double doubleValueFromInt = integerFromInt.doubleValue();  
  
byte byteValueFromString = integerFromString.byteValue();  
int intValueFromString = integerFromString.intValue();  
String stringValue = integerFromString.toString();  
double doubleValueFromString = integerFromString.doubleValue();
```

Parsowanie

- To mechanizm podobny do rzutowania. Służy do zamiany klasy String na inne typy proste. Do odwrotnej operacji należy użyć statycznej metody „valueOf” z klasy String

```
int intValue = Integer.parseInt( s: "123");  
double doubleValue = Double.parseDouble( s: "100.233");  
boolean booleanValue = Boolean.parseBoolean( s: "true");
```

```
String stringInt = String.valueOf(123);  
String stringDouble = String.valueOf(342.233);  
String stringBoolean = String.valueOf(true);
```

Klasa Date



Domyślnym obiektem do reprezentacji daty jest `Date`. Przechowuje ona informacje w formie liczby milisekund które upłynęły od dnia 01.01.1970.

Konstruktor klasy `Date` przyjmuje wartość typu „long”. Po użyciu konstruktora bezargumentowego zostanie przyjętą aktualna data.

Project

1: Project

Demo

C:\Users\dobne\IdeaProjects\Demo

> .idea

> out

> src

> com.company

> Main

Demo.iml

> External Libraries

> Scratches and Consoles

Main.java

```
1 package com.company;
2
3 import java.util.Date;
4
5 public class Main {
6
7     public static void main(String[] args) {
8
9         Date today = new Date();
10        Date theOldestDate = new Date(0);
11        Date maxDate = new Date(Long.MAX_VALUE);
12
13        System.out.println("Today is " + today);
14        System.out.println("First date is " + theOldestDate);
15        System.out.println("Max date is " + maxDate);
16
17    }
18 }
```

Run: Main

Run

↑

↓

Copy

Paste

Close

Refresh

Search

2: Favorites

>>

```
"C:\Program Files\Java\jdk-11.0.8\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2020.2.2\lib\idea_rt.jar=51
Today is Thu Jul 22 23:34:44 CEST 2021
First date is Thu Jan 01 01:00:00 CET 1970
Max date is Sun Aug 17 08:12:55 CET 292278994

Process finished with exit code 0
```

Metody klasy Date

- `date.after(date2)` – sprawdza czy data w zmiennej „date” występuje po zmiennej „date1”.
- `date.before(date2)` – sprawdza czy data w zmiennej „date” występuje przed zmienną „date1”.
- `date.compareTo(date2)` – porównuje daty, jeżeli „date” występuje przed „date2” to metoda zwróci -1, jeżeli po 1 a jeżeli daty są równe wynikiem będzie 0.
- `date.getTime()` – zwraca wartość w formie liczby milisekund od daty 01.01.1970.

Formatowanie i parsowanie daty

Aby zmienić formatowanie wyświetlanej daty należy użyć klasy do jej formatowania. Domyślnie w Javie są to pochodne klasy od `DateFormat`

```
DateFormat dateFormat = new SimpleDateFormat( pattern: "yyyy-MM-dd HH:mm:ss");
```

Wzór

dd-MM-yy

dd-MM-yyyy

MM-dd-yyyy

yyyy-MM-dd

yyyy-MM-dd HH:mm:ss

yyyy-MM-dd HH:mm:ss.SSS

yyyy-MM-dd HH:mm:ss.SSSZ

EEEE MMMMM yyyy HH:mm:ss.SSSZ

Przykład

31-01-12

31-01-2012

01-31-2012

2012-01-31

2012-01-31 23:59:59

2012-01-31 23:59:59.999

2012-01-31 23:59:59.999+0100

Saturday November 2012 10:45:42.720+0100

Project

1: Project

Demo

C:\Users\dobne\IdeaProjects\Demo

.idea

out

src

com.company

Main

Demo.iml

External Libraries

Scratches and Consoles

Main.java

```

5  import java.text.SimpleDateFormat;
6  import java.util.Date;
7
8  public class Main {
9
10     public static void main(String[] args) {
11
12         Date today = new Date();
13         DateFormat dateFormat = new SimpleDateFormat( pattern: "yyyy-MM-dd HH:mm:ss");
14         System.out.println("Today is " + dateFormat.format(today));
15
16         try {
17             Date otherDate = dateFormat.parse( source: "2141-04-11 05:22:17");
18             System.out.println("Other date is " + otherDate);
19         } catch (ParseException e) {
20             System.out.println("Problem with parsing date.");
21             e.printStackTrace();
22         }
23     }
24

```

Run: Main

Run

Up

Down

Copy

Paste

Print

Close

```

"C:\Program Files\Java\jdk-11.0.8\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2020.2.2\lib\idea_rt.jar=51
Today is 2021-07-23 00:22:02
Other date is Tue Apr 11 05:22:17 CEST 2141

Process finished with exit code 0

```

4: Run TODO 6: Problems Terminal Build Event Log

Build completed successfully in 9 s 639 ms (2 minutes ago) 24:2 CRLF UTF-8 4 spaces

Klasa Calendar

- Klasa Calendar upraszcza operacje na datach.
- Do zainicjalizowania zmiennej wykorzystywana jest statyczna metoda „getInstance()”. Nie da się powołać obiektu Calendar przy użyciu konstruktora.



Project

Demo

.idea

out

src

com.company

Main

Demo.iml

External Libraries

Scratches and Consoles

Main.java

11 public static void main(String[] args) {
12
13 Calendar calendar = Calendar.getInstance();
14 System.out.println(calendar.getTime());
15
16 calendar.set(year: 0, month: 0, date: 1);
17 System.out.println(calendar.getTime());
18
19 calendar.set(year: 1, month: 0, date: 1);
20 System.out.println(calendar.getTime());
21
22 calendar.set(year: 2012, month: 12, date: 1, hourOfDay: 12, minute: 50, second: 59);
23 System.out.println(calendar.getTime());
24
25 calendar.set(year: 2012, month: 1, date: 29);
26 System.out.println(calendar.getTime());
27
28 calendar.set(year: 2012, month: 1, date: 30);
29 System.out.println(calendar.getTime());
30
31 }

Run: Main

C:\Program Files\Java\jdk-11.0.6\bin\java.exe -javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2020.2.2\lib\idea_11.0.jar -30

Fri Aug 06 21:56:20 CEST 2021
Thu Jan 01 21:56:20 CET 1
Thu Jan 01 21:56:20 CET 1
Fri Jan 01 12:50:59 CET 2011
Sun Mar 01 12:50:59 CET 2012
Mon Mar 02 12:50:59 CET 2012

Process finished with exit code 0

Pola kalendarza

ERA: int = 0
YEAR: int = 1
MONTH: int = 2
WEEK_OF_YEAR: int = 3
WEEK_OF_MONTH: int = 4
DATE: int = 5
DAY_OF_MONTH: int = 5
DAY_OF_YEAR: int = 6
DAY_OF_WEEK: int = 7
DAY_OF_WEEK_IN_MONTH: int = 8
AM_PM: int = 9
HOUR: int = 10
HOUR_OF_DAY: int = 11
MINUTE: int = 12
SECOND: int = 13
MILLISECOND: int = 14
ZONE_OFFSET: int = 15
DST_OFFSET: int = 16
FIELD_COUNT: int = 17
SUNDAY: int = 1
MONDAY: int = 2

TUESDAY: int = 3
WEDNESDAY: int = 4
THURSDAY: int = 5
FRIDAY: int = 6
SATURDAY: int = 7
JANUARY: int = 0
FEBRUARY: int = 1
MARCH: int = 2
APRIL: int = 3
MAY: int = 4
JUNE: int = 5
JULY: int = 6
AUGUST: int = 7
SEPTEMBER: int = 8
OCTOBER: int = 9
NOVEMBER: int = 10
DECEMBER: int = 11
UNDECIMBER: int = 12
AM: int = 0
PM: int = 1
ALL_STYLES: int = 0
STANDALONE_MASK: int = 0x8000
SHORT: int = 1
LONG: int = 2

- Za pomocą nazwy odpowiedniego pola i metody „get()” możemy pobrać konkretną wartość z kalendarza, jak i specjalną metodą „set()” ustawić tylko jedno pole jednocześnie.

Project

1: Project

Demo C:\Users\dobne\IdeaProjects\Demo

.idea

out

src

com.company

Main

Demo.iml

External Libraries

Scratches and Consoles

Main.java x

```

6
7 public static void main(String[] args) {
8
9     Calendar calendar = Calendar.getInstance();
10    System.out.print("Today is ");
11    System.out.println(calendar.getTime());
12    System.out.println("Year is " + calendar.get(Calendar.YEAR));
13    System.out.println("Month is " + calendar.get(Calendar.MONTH));
14    System.out.println("Day of year is " + calendar.get(Calendar.DAY_OF_YEAR));
15    System.out.println("Week of month is " + calendar.get(Calendar.WEEK_OF_MONTH));
16
17    calendar.set(Calendar.YEAR, 2030);
18    System.out.println(calendar.getTime());
19    calendar.set(Calendar.DATE, 15);
20    System.out.println(calendar.getTime());
21    calendar.set(Calendar.DAY_OF_WEEK, Calendar.SATURDAY);
22    System.out.println(calendar.getTime());
23    calendar.set(Calendar.MONTH, Calendar.JANUARY);
24    System.out.println(calendar.getTime());
25

```

Run: Main x

Run

↑

↓

↺

↻

⌂

🔍

📄

🗑️

```

Today is Fri Aug 06 22:20:00 CEST 2021
Year is 2021
Month is 7
Day of year is 218
Week of month is 1
Tue Aug 06 22:20:00 CEST 2030
Thu Aug 15 22:20:00 CEST 2030
Sat Aug 17 22:20:00 CEST 2030
Thu Jan 17 22:20:00 CET 2030

```

Operacje na datach

- W celu zmiany daty o jakąś wartość najlepiej posłużyć się metodą „add()” pamiętając, że wartości ujemne w parametrze będą powodować odejmowanie od danego pola.

```
Calendar calendar = Calendar.getInstance();  
calendar.add(Calendar.MONTH, amount: 2);  
calendar.add(Calendar.DATE, amount: -12);  
calendar.add(Calendar.HOUR, amount: 5);
```

Calendar calendar = Calendar.getInstance();	→	Fri Aug 06 22:27:26 CEST 2021
calendar.add(Calendar.MONTH, amount: 2);	→	Wed Oct 06 22:27:26 CEST 2021
calendar.add(Calendar.DATE, amount: -12);	→	Fri Sep 24 22:27:26 CEST 2021
calendar.add(Calendar.HOUR, amount: 5);	→	Sat Sep 25 03:27:26 CEST 2021

Koniec ;) Czas na proste ćwiczenia

