3. Essential Java Classes

3. Date and Time

Date and Time Classes

- **Date** - represents a specific instant in time, with millisecond precision
- **GregorianCalendar** (concrete subclass of **Calendar**) – date and time manipulations
- **SimpleDateFormat**- formatting and parsing dates in a locale-sensitive manner
- **Locale** - represents a specific geographical, political, or cultural region
- **SimpleTimeZone**(concrete subclass of **TimeZone**) - represents a time zone for use with a Gregorian calendar

Class Date

- represents a specific instant in time, with millisecond precision
- new Date() returns current date and time
- Date saves the milliseconds since January 1, 1970, 00:00:00
- Date does not localized. Most of it methods are deprecated.
- Calendar class allows to process dates

Some Date methods

- equals(Object obj) - compares dates
- after(Date date) - compares dates
- before(Date date) - compares dates
- getTime() – returns milliseconds since 01.01.1970
- setTime() – sets date and time

See http://docs.oracle.com/javase/7/docs/api/java/util/Date.html for details

Class Calendar

- Abstract class that provides methods for converting between a specific instant in time and a set of calendar fields
- Some calendar fields: YEAR, MONTH, DATE, DAY_OF_WEEK, HOUR_OF_DAY, MINUTE, SECOND, MILLISECOND

Calendar Methods

- **Calendar rightNow = Calendar.getInstance();** - gets current date and time
- **set(int field, int value)** – sets calendar field value (then one of **get (), getTime(), add(), roll()** methods is needed)
- **add(int field, int amount)** – adds given amount to the field
- **get(int field)** – gets a given field value

See http://docs.oracle.com/javase/7/docs/api/java/util/Calendar.html for details

Class GregorianCalendar

- Concrete subclass of Calendar
- Provides the standard calendar system used by most of the world.
- Use GregorianCalendar for date and time manipulations

See http://docs.oracle.com/javase/7/docs/api/java/util/GregorianCalendar.html for details

Calendar Examples

- Задание даты:

```
GregorianCalendar calendar = new GregorianCalendar(2012,
    Calendar.May, 14);
```

- Добавление к дате двух недель:

```
calendar.add(Calendar.WEEK_OF_YEAR, 2);
```

- Изменение полей даты:

```
calendar.set(Calendar.DAY_OF_MONTH, 10);
calendar.set(Calendar.MONTH, Calendar.SEPTEMBER);
System.out.println(dtFrm.format(calendar.getTime()));
```

Date & Calendar Transformations

Calendar c = Calendar.*getInstance();*

Date dt = new Date();

- Calendar -> Date

  dt = c.getTime();

- Date -> Calendar

  c.setTime(dt);

Class SimpleDateFormat

- is a concrete class for formatting and parsing dates in a locale-sensitive manner
- It allows for:
  - formatting (date -> text)
  - parsing (text -> date)
- See http://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html for details

SimpleDateFormat Example

- SimpleDateFormat dtFrm =
  new SimpleDateFormat("dd.MM.yyyy");
- Date start = new Date();
- String txDate = dtFrm.format(start);

Class Locale

- represents a specific geographical, political, or cultural region
- Locale lc = new Locale(”uk”, ”UK”);
- Locale.getDefault() - gets the current value of the default locale

See http://docs.oracle.com/javase/7/docs/api/java/util/Locale.html for details

Класс SimpleTimeZone

- A concrete subclass of TimeZone class that represents a time zone for use with a Gregorian calendar
- Can specify the year when the daylight saving time schedule starts or ends.

See http://docs.oracle.com/javase/7/docs/api/java/util/SimpleTimeZone.html for details

SimpleDateFormat Localization

```
SimpleDateFormat dtFrmL = new
      SimpleDateFormat("dd.MMMM.yyyy", Locale.ENGLISH);
Date dt = new Date();
System.out.println("Locale.ENGLISH: " + dtFrmL.format(dt));
```

Locale.ENGLISH: 05.October.2013
Locale.GERMAN: 05.Oktober.2013
Locale.FRANCE: 05.octobre.2013
Locale.ITALIAN: 05.ottobre.2013
Locale("ua", "ua"): 05.October.2013
Locale("ru", "ru"): 05.Октябрь.2013

DateFormat Localization

```
DateFormat dtFrmD =
DateFormat.getDateInstance(DateFormat.DEFAULT, Locale.ENGLISH);
System.out.println("Locale.ENGLISH: " + dtFrmD.format(dt));
```

Locale.ENGLISH: Oct 5, 2013

Locale.GERMAN: 05.10.2013

Locale.FRANCE: 5 oct. 2013

Locale.ITALIAN: 5-ott-2013

Locale("ua", "ua"): Saturday, October 5, 2013

Locale("ru", "ru"): 5 Октябрь 2013 г.

Date Localization

- http://docs.oracle.com/javase/tutorial/i18n/format/dateFormat.html

Exercise 3.3.1

- Print the following:
  - Current date
  - Date in 6 weeks
  - Date 4 month before
  - Date in 45 days

Exercise 3.3.1

- See 331DateActions project for the full text

Exercise 3.3.2.

- Create a static method that gets some date and returns next bank day

Exercise 3.3.2.

- See 332NextBankDay project for full text

Exercise 3.3.3.

- Create a method that gets two dates and returns number of days between these dates (general idea only!!!).