sages

Course code: **J/OOP**

Course title: **Programming and object-oriented programming – the example of Java language**

Days: 3

Description:

Course intended for:

The training is an introduction to object-based languages, using Java as an example. The participants will get familiar with the basic concepts of objectivity (inheritance, polymorphism etc.). Emphasis is put mainly on the practical skills associated with use of the proper object-oriented techniques in the context of specific problems.

Course objective:

After the training, the participants should have the following knowledge/ skills:

- Effective use of object-oriented concepts – classes, abstract classes, inheritance, polymorphism etc.
- Model problems using the above concepts
- Identify design problems and appropriately select the tools needed
- Get basic knowledge of design patterns and understand the reasons for their use

Requirements:

The training can be attended by programmers, starting to learn object-oriented programming languages, such as Java, C#. Although the training is based on Java language, its structure allows for easy transferring of the concepts learned to other object-oriented languages.

Course parameters:

3*8 hours (3*7 net hours) of lectures and workshops (with a visible emphasis on workshops).

Group size: no more than 8-10 participants.

Course curriculum:

Sages sp. z o.o., ul. Nowogrodzka 62c, 02-002 Warszawa
tel. +48 22 203 56 00 fax: +48 22 203 56 01
e-mail: office@sages.io web: www.sages.io

sages

1. Introduction
    I. Imperative programming
    II. Declarative programming
    III. Object-oriented programming
    IV. Functional programming
2. Installation and configuration of the work environment (IntelliJ IDEA, Eclipse IDE or NetBeans IDE depending on group preferences)
3. Object orientation – the example of Java language: basic topics:
    I. Classes and objects
    II. Interfaces
    III. Enumerations
    IV. Abstract classes
    V. Inheritance
4. Object orientation – the example of Java language: advanced topics:
    I. Polymorphism
    II. Internal classes
    III. Anonymous classes
5. Modeling of problems using UML
    I. Introduction to UML notation
    II. The class diagram
        i. Associations
        ii. Aggregations
        iii. Dependencies

Sages sp. z o.o., ul. Nowogrodzka 62c, 02-002 Warszawa
tel. +48 22 203 56 00 fax: +48 22 203 56 01
e-mail: office@sages.io web: www.sages.io

iv. Generalizations

v. realizations

6. Object-oriented programming- a practical approach

I. The principle of responsibility

II. Why encapsulation is (usually) better than multi-level inheritance?

III. Programming through interfaces

IV. A code defending itself against modification, encouraging development

7. Introduction to design patterns

I. Gang-of-Four (GoF)

II. Singleton as an example of a creational pattern

III. Decorator as an example of a structural pattern

IV. Template as an example of a behavioral pattern