IPT – Intellectual Products & Technologies Ltd.
Bl. 44- Office 2 Slatinska Str., Sofia, Bulgaria
☏ (+359-2) 973-38-38 🖷 (+359-2) 971-28-50 ✉ office@iproduct.org 🖱 http://www.iproduct.org/

# *PROGRAMMING EXERCISES*

***Problem 1: Program for invoicing***

a) Write a program for invoicing, working in text mode:

b) Create a class **Product** with attributes: 1) unique identifier of the goods; 2) goods code (up to 10 letters and digits); 3) name of the goods; and 4) price of goods; and methods: 1) constructor without arguments (default constructor); 2) constructor with three arguments (code, name and price of the goods); 3) method **toString()**, which returns information about the goods in text; 4) method **input()**, which to input the attributes of the goods from the keyboard. Create also a static method **main**, that creates three different products (goods), the third is entered from the keyboard and these products are printed on the screen in text.

c) Create a class **Contragent** with attributes: 1) identification number; 2) name; 3) address; 4) individual or company; 5) telephone. Create following methods: 1) constructor without arguments (default constructor); 2) constructor with 4 arguments – mandatory attributes; 3) method **toString()**, which returns information about contractor in text, 4) method **input()**, which to input the attributes of new contractor from the keyboard. Create also a static method **main**, which creates three different contractors, the third is entered from the keyboard and the data for these contractors are printed on the screen in text.

d) Create a class **Invoice** with attributes: 1) unique invoice number; 2) date of issue; 3) date of payment or performance of the transaction; 4) person issuing the invoice; 5) person received the invoice; and 6) one or more positions. Each position (class **Position**) has: 1) item number, 2) unique identifier of the goods, 3) quantity; 4) unit of measure (kg, pcs., etc.). Implement following methods: 1) constructor without arguments (default constructor), 2) constructor with four arguments (number, date, person received the invoice, list of items), 3) constructor with all arguments (full constructor); 4) method **toString()**, which returns information about the receipt in text. Create also a static method **main**, to create a new invoice and print it on the screen in text.

e) Create a class **InvoiceRegister** with attributes you decide and basic methods:

1. ***initialize()*** - initialize the system by creating a list of three products and three contractors (maybe from the same subsection 1.) initialize the data of the company issuing the first contractor;
2. ***newInvoice(...)***, which from existing **positions list** and identification number of the recipient type **Contragent** creates object from class **Invoice** and return it as a result;
3. ***printInvoice(...)***, which from argument object type **Invoice**, prints invoice on screen, including information about the company-publisher of the invoice and the recipient, number and date of invoice, items list, with displayed the names of the goods and the value of each position (quantity multiplied by the price of the product), and the total amount, VAT and amount including VAT.

4. ***main(...)*** – static method that prints an invoice with three positions (quantities are 1, 5 and 10) on the console.

***Problem 2: Invoicing - separate packages***

a) Move classes **Product, Contragent, Position** and **Invoice** in package **invoicing.model**

b) Move class **InvoiceRegister** in package **invoicing.controller**

c) Set the appropriate ***import*** structures and specifiers to access various classes and their methods.

***Problem 3: Program to issue invoices - factoring in inheritance***

Modify the program from Problem 1, so that the class **Contragent** to be extended from class **Company** with additional attributes: 6) VAT registration; 7) responsible person; 8) BIC; 9) IBAN. Create following methods in the new class: 1) constructor without arguments (default constructor); 2) constructor with 4 argument - only with mandatory attributes from the base class **Contragent**; 3) constructor with all 9 arguments (full constructor); 3) method **toString()**, which returns complete information about the company in text, 4) method **input()**, which to input the attributes of new company from the keyboard. Modify classes **Invoice** and **InvoiceRegister**, so that the issuer of the invoice can be only company. Modify also static method **main** of the class **InvoiceRegister**, so to print two invoices - first with set from the keyboard recipient - company and the second with set from the keyboard recipient - individual. Invoices have three positions (quantities are 1, 5 and 10).

## Literature and Internet Resources

1. Oracle® Java™ Technologies webpage – http://www.oracle.com/technetwork/java/index.html
2. Eckel, B., Thinking in Java. 4-th ed., Prentice Hall, 2006 – http://mindview.net/Books/TIJ4
3. Effective Java Second Edition, Bloch, J., Sun Microsystems, 2008
4. Schildt, H., Java 2 - Developer Guide. Softpress, in bg, 2007
5. Eck, D., Introduction to Programming Using Java, Fifth Edition, Version 5.1, June 2009 – http://math.hws.edu/javanotes/