МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний аерокосмічний університет ім. М.Є. Жуковського “ХАІ”

Кафедра 603

**Лабораторна робота № 2**

З дисципліни “Програмування для мережі Інтернет”

# Модулювання об’єктів за допомогою мови Java

ХАІ.603.631п.15В.050103.126335.ПЗ

Виконав студент гр. 631п Кошиль В. А.

(№ групи) (П.І.Б)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис, дата)

Перевірив:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(наукова ступінь, вчене звання)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис, дата) (П.І.Б)

Харків, 2015

# Лабораторная работа № 2

# Моделирование объектов средствами языка Java

## Цель работы:

Получение практических навыков по применению классов, интерфейсов и методов в программных приложениях на языке Java.

## Изучаемый материал:

Ссылочные типы данных языка Java: назначение, объявление, структура и особенности применения. Модификаторы доступа полей и методов класса. Наследование классов и интерфейсов.

# Постановка задачи

Разработать и реализовать на языке Java приложение для моделирования свойств, поведения и взаимодействия объектов. Предметной областью данного приложения могут служить любые объекты реального мира. В приложении продемонстрировать применение абстрактных классов, интерфейсов, механизмов наследования и полиморфизма.

**Вариант 10**

Транспортные средства (автобусы, трамваи, троллейбусы). Свойства: тип модели, фирма-производитель, номер, маршрут, вместимость, закрепленный водитель и т. п.. Поведение: определение и изменение свойств, определение состояния и т. п.

# Описание классов и методов

Во время выполнения лабораторной работы были спроектированы 4 класса:

* Transport – базовый класс реализующий интерфейс Transport
* Bus – класс, представляющий объект «автобус».
* Trolley – класс, представляющий объект «троллейбус».
* Tram – класс, представляющий объект «трамвай».

## Класс Drink

|  |
| --- |
| **Класс Transport** |
| **Методы** |
| + GetType():String |
| + SetType(String type):void |
| + GetFirm():String |
| + SetFirm(String Name):void |
| + GetNumber():int |
| + SetNumber(int number):void |
| + GetWay(): String |
| + SetWay(String way):void |
| + GetCapacity():int |
| + SetCapacity(int capacity) |
| + GetDriver():String |
| + SetDriver(String driver) |

Таблица 1 – Диаграмма класс Transport

## Класс Bus

|  |
| --- |
| **Класс Bus** |
| **Методы** |
| + GetType():String |
| + SetType(String type):void |
| + GetFirm():String |
| + SetFirm(String Name):void |
| + GetNumber():int |
| + SetNumber(int number):void |
| + GetWay(): String |
| + SetWay(String way):void |
| + GetCapacity():int |
| + SetCapacity(int capacity) |
| + GetDriver():String |
| + SetDriver(String driver) |
| -type |
| -firm |
| -number |
| -way |
| -capacity |
| -driver |

Таблица 2 – Диаграмма класса Bus

## Класс Trolley

|  |
| --- |
| **Класс Trolley** |
| **Методы** |
| + GetType():String |
| + SetType(String type):void |
| + GetFirm():String |
| + SetFirm(String Name):void |
| + GetNumber():int |
| + SetNumber(int number):void |
| + GetWay(): String |
| + SetWay(String way):void |
| + GetCapacity():int |
| + SetCapacity(int capacity) |
| + GetDriver():String |
| + SetDriver(String driver) |
| -type |
| -firm |
| -number |
| -way |
| -capacity |
| -driver |

Таблица 3 – Диаграмма класса Trolley

## Класс Tram

|  |
| --- |
| **Класс Tram** |
| **Методы** |
| + GetType():String |
| + SetType(String type):void |
| + GetFirm():String |
| + SetFirm(String Name):void |
| + GetNumber():int |
| + SetNumber(int number):void |
| + GetWay(): String |
| + SetWay(String way):void |
| + GetCapacity():int |
| + SetCapacity(int capacity) |
| + GetDriver():String |
| + SetDriver(String driver) |
| -type |
| -firm |
| -number |
| -way |
| -capacity |
| -driver |

Таблица 3 – Диаграмма класса Tram

# Выводы

Во время выполнения лабораторной работы были изучены и использованы в работе ссылочные типы данных языка Java: назначение, объявление, структура и особенности применения. Также были изучены модификаторы доступа полей и методов класса. Наследование классов и интерфейсов. Также были реализованы интерфейс Transport и классы Bus, Trolley, Tram.

# Листинг

Файл Transport.java

**public** **interface** Transport {

**public** String GetType();

**public** **void** SetType(String type);

**public** String GetFirm();

**public** **void** SetFirm(String firm);

**public** **int** GetNumber();

**public** **void** SetNumber(**int** number);

**public** String GetWay();

**public** **void** SetWay(String way);

**public** **int** GetCapacity();

**public** **void** SetCapacity(**int** capacity);

**public** String GetDriver();

**public** **void** SetDriver(String driver);

}

Файл Bus.java

**public** **class** Bus **implements** Transport {

**public** String GetType() {

**return** **this**.type;

}

**public** **void** SetType(String type) {

**this**.type = type;

}

**public** String GetFirm() {

**return** **this**.firm;

}

**public** **void** SetFirm(String firm) {

**this**.firm = firm;

}

**public** **int** GetNumber(){

**return** **this**.number;

}

**public** **void** SetNumber(**int** number){

**this**.number=number;

}

**public** String GetWay() {

**return** **this**.way;

}

**public** **void** SetWay(String way){

**this**.way=way;

}

**public** **int** GetCapacity(){

**return** **this**.capacity;

}

**public** **void** SetCapacity(**int** capacity){

**this**.capacity=capacity;

}

**public** String GetDriver(){

**return** **this**.driver;

}

**public** **void** SetDriver(String driver){

**this**.driver=driver;

}

**private** String type;

**private** String firm;

**private** **int** number;

**private** String way;

**private** **int** capacity;

**private** String driver;

}

Файл Trolley.java

**public** **class** Trolley **implements** Transport {

**public** String GetType() {

**return** **this**.type;

}

**public** **void** SetType(String type) {

**this**.type = type;

}

**public** String GetFirm() {

**return** **this**.firm;

}

**public** **void** SetFirm(String firm) {

**this**.firm = firm;

}

**public** **int** GetNumber(){

**return** **this**.number;

}

**public** **void** SetNumber(**int** number){

**this**.number=number;

}

**public** String GetWay() {

**return** **this**.way;

}

**public** **void** SetWay(String way){

**this**.way=way;

}

**public** **int** GetCapacity(){

**return** **this**.capacity;

}

**public** **void** SetCapacity(**int** capacity){

**this**.capacity=capacity;

}

**public** String GetDriver(){

**return** **this**.driver;

}

**public** **void** SetDriver(String driver){

**this**.driver=driver;

}

**private** String type;

**private** String firm;

**private** **int** number;

**private** String way;

**private** **int** capacity;

**private** String driver;

}

Файл Tram.java

**public** **class** Tram **implements** Transport {

**public** String GetType() {

**return** **this**.type;

}

**public** **void** SetType(String type) {

**this**.type = type;

}

**public** String GetFirm() {

**return** **this**.firm;

}

**public** **void** SetFirm(String firm) {

**this**.firm = firm;

}

**public** **int** GetNumber(){

**return** **this**.number;

}

**public** **void** SetNumber(**int** number){

**this**.number=number;

}

**public** String GetWay() {

**return** **this**.way;

}

**public** **void** SetWay(String way){

**this**.way=way;

}

**public** **int** GetCapacity(){

**return** **this**.capacity;

}

**public** **void** SetCapacity(**int** capacity){

**this**.capacity=capacity;

}

**public** String GetDriver(){

**return** **this**.driver;

}

**public** **void** SetDriver(String driver){

**this**.driver=driver;

}

**private** String type;

**private** String firm;

**private** **int** number;

**private** String way;

**private** **int** capacity;

**private** String driver;

}

Файл Main.java

**import** java.util.Scanner;

**import** java.util.Vector;

**public** **class** MainClass {

**public** **static** **void** main(String[] args) {

Scanner scan = **new** Scanner(System.***in***);

Vector<Transport> transports = **new** Vector<Transport>();

Transport transport;

**int** choose=0;

**while**(**true**){

System.***out***.println("\nChose the action:");

System.***out***.println("1 - Put bus on the route");

System.***out***.println("2 - Put trolley on the route");

System.***out***.println("3 - Put tram on the route");

System.***out***.println("4 - Show autopark");

System.***out***.println("5 - Remove car from autopark");

System.***out***.println("6 - Destroy the autopark");

System.***out***.println("7 - Exit");

choose = scan.nextInt();

**switch**(choose){

**case** 1:

transport = **new** Bus();

System.***out***.println("Input firm: ");

transport.SetFirm(scan.next());

System.***out***.println("Input type: ");

transport.SetType(scan.next());

System.***out***.println("Input number: ");

transport.SetNumber(scan.nextInt());

System.***out***.println("Input way: ");

transport.SetWay(scan.next());

System.***out***.println("Input capacity: ");

transport.SetCapacity(scan.nextInt());

System.***out***.println("Input driver: ");

transport.SetDriver(scan.next());

transports.add(transport);

System.***out***.println("Done! Bus is put on course.");

**break**;

**case** 2:

transport = **new** Trolley();

System.***out***.println("Input firm: ");

transport.SetFirm(scan.next());

System.***out***.println("Input type: ");

transport.SetType(scan.next());

System.***out***.println("Input number: ");

transport.SetNumber(scan.nextInt());

System.***out***.println("Input way: ");

transport.SetWay(scan.next());

System.***out***.println("Input capacity: ");

transport.SetCapacity(scan.nextInt());

System.***out***.println("Input driver: ");

transport.SetDriver(scan.next());

transports.add(transport);

System.***out***.println("Done! Trolley is put on course.");

**break**;

**case** 3:

transport = **new** Tram();

System.***out***.println("Input firm: ");

transport.SetFirm(scan.next());

System.***out***.println("Input type: ");

transport.SetType(scan.next());

System.***out***.println("Input number: ");

transport.SetNumber(scan.nextInt());

System.***out***.println("Input way: ");

transport.SetWay(scan.next());

System.***out***.println("Input capacity: ");

transport.SetCapacity(scan.nextInt());

System.***out***.println("Input driver: ");

transport.SetDriver(scan.next());

transports.add(transport);

System.***out***.println("Done! Tram is put on course.");

**break**;

**case** 4:

**for**(**int** i=0; i<transports.size(); i++){

System.***out***.println("The "+(i+1)+" car in autopark: ");

System.***out***.println("Car made by: " + transports.elementAt(i).GetFirm());

System.***out***.println("Car type is: " + transports.elementAt(i).GetType());

System.***out***.println("Car number is: " + transports.elementAt(i).GetNumber());

System.***out***.println("Car course is: " + transports.elementAt(i).GetWay());

System.***out***.println("Car capacity is: " + transports.elementAt(i).GetCapacity());

System.***out***.println("Driver name is: " + transports.elementAt(i).GetDriver());

}

**break**;

**case** 5:

System.***out***.println("Input position will be removed!");

transports.remove(scan.nextInt() - 1);

**break**;

**case** 6:

System.***out***.println("Autopark will be destroyed!");

transports.clear();

**break**;

**case** 7:

System.*exit*(0);

}

}

}

}