Частное учреждение образования

«Колледж бизнеса и права»

Тема: «Наследование и инкапсуляция в языке Java»

Учащийся М.Н. Гончаров.

группа: Т-992

2021

Текст задачи 1

1. Создайте проект Java. Назовите пакет com.example , а главный класс EmployeeTest

2. Создайте пакет com.example.domain, а в нем класс Employee с указанными полями: public int empId; public String name; public String ssn; public double salary;

3. Добавьте конструктор класса: public Employee() {}

4. Создате методы чтения и записи ( «геттеры» и «сеттеры») для каждого поля. Используйте для этого контекстное меню редактора NetBeans

5. Добавьте в файл класса EmployeeTest импорт класса Employee import com.example.domain.Employee;

6. Добавьте в процедуру main класса EmployeeTest команды создания объекта класса 7 Employee и заполнение его полей Employee emp = new Employee(); emp.setEmpId(101); emp.setName("Jane Smith"); emp.setSalary(120\_345.27); emp.setSsn("012-34-5678");

7. Добавьте в процедуру main класса EmployeeTest команды отображения данных объекта класса Employee System.out.println("Employee ID: "+emp.getEmpId()); System.out.println("Employee Name: "+emp.getName()); System.out.println("Employee Soc Sec # "+emp.getSsn()); System.out.println("Employee salary: "+emp.getSalary());

8. Запустите приложение.

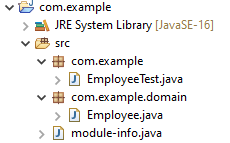


Рисунок 1

// pakage - com.example – EmployeeTest.java

**package** com.example;

**import** com.example.domain.Employee;

**public** **class** EmployeeTest {

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

Employee emp = **new** Employee();

emp.setEmpId(101);

emp.setName("Jane Smith");

emp.setSalary(120\_345.27);

emp.setSsn("012-34-5678");

System.***out***.println("Employee ID: "+emp.getEmpId());

System.***out***.println("Employee Name: "+emp.getName());

System.***out***.println("Employee Soc Sec # "+emp.getSsn());

System.***out***.println("Employee salary: "+emp.getSalary());

}

}

// pakage - com.example.domain – Employee.java

**package** com.example.domain;

**public** **class** Employee {

**public** **int** empId;

**public** String name;

**public** String ssn;

**public** **double** salary;

// Конструктор без параметров

**public** Employee() {}

// get

**public** **int** getEmpId()

{

**return** empId;

}

**public** String getName()

{

**return** name;

}

**public** String getSsn()

{

**return** ssn;

}

**public** **double** getSalary()

{

**return** salary;

}

// set

**public** **void** setEmpId(**int** x)

{

empId = x;

}

**public** **void** setName(String x)

{

name = x;

}

**public** **void** setSsn(String x)

{

ssn = x;

}

**public** **void** setSalary(**double** x)

{

salary = x;

}

}

Результат работы программы представлен на рисунке 2

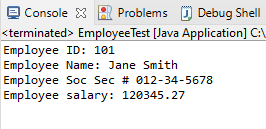


Рисунок 2

Текст задачи 2

Создайте следующие классы по вариантам:

Вариант 6. Программное обеспечение, 4 поля– 2 класса наследника

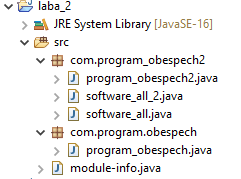


Рисунок 3

// pakage - com.program\_obespech2 – program\_obespech2.java

**package** com.program\_obespech2;

// Класс родитель

**public** **class** program\_obespech2 {

**private** **int** pole1;

**private** String name;

**public** program\_obespech2(){};

**public** program\_obespech2(**int** pole1, String name)

{

**this**.pole1 = pole1;

**this**.name = name;

}

**public** **int** getPole1() {

**return** pole1;

}

**public** **void** setPole1(**int** pole1) {

**this**.pole1 = pole1;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

}

// pakage - com.program\_obespech2 – software\_all\_2.java

**package** com.program\_obespech2;

**public** **class** software\_all\_2 **extends** program\_obespech2 {

**int** flag2;

**public** software\_all\_2(){};

**public** software\_all\_2(**int** pole1, String name, **int** flag2)

{

// Вызываю конструктор класса родителя

**super**(pole1, name);

**this**.flag2 = flag2;

}

**public** **int** getFlag2() {

**return** flag2;

}

**public** **void** setFlag2(**int** flag2) {

**this**.flag2 = flag2;

}

}

// pakage - com.program\_obespech2 – software\_all.java

**package** com.program\_obespech2;

**public** **class** software\_all **extends** program\_obespech2{

**int** flag1;

**public** software\_all(){};

**public** software\_all(**int** pole1, String name, **int** flag1)

{

// Вызываю конструктор класса родителя

**super**(pole1, name);

**this**.flag1 = flag1;

}

**public** **int** getFlag1() {

**return** flag1;

}

**public** **void** setFlag1(**int** flag1) {

**this**.flag1 = flag1;

}

}

// pakage - com.program\_obespech – program\_obespech.java

**package** com.program.obespech;

**import** com.program\_obespech2.program\_obespech2;

**import** com.program\_obespech2.software\_all;

**import** com.program\_obespech2.software\_all\_2;

**public** **class** program\_obespech {

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

// Родитель

program\_obespech2 obj0 = **new** program\_obespech2(1, "test");

// Потомки

software\_all obj1 = **new** software\_all(20, "free", 77);

software\_all\_2 obj2 = **new** software\_all\_2(10, "non free", 88);

System.***out***.println("obj0 - pole1 = " + obj0.getPole1());

System.***out***.println("obj0 - name = " + obj0.getName());

System.***out***.println("\nobj1 - pole1 = " + obj1.getPole1());

System.***out***.println("obj1 - name = " + obj1.getName());

System.***out***.println("obj1 - flag1 = " + obj1.getFlag1());

obj1.setPole1(100);

System.***out***.println("\nobj1 - pole1 = " + obj1.getPole1());

System.***out***.println("\nobj2 - pole1 = " + obj2.getPole1());

System.***out***.println("obj2 - name = " + obj2.getName());

System.***out***.println("obj2 - flag2 = " + obj2.getFlag2());

}

}

Результат работы программы представлен на рисунке 4

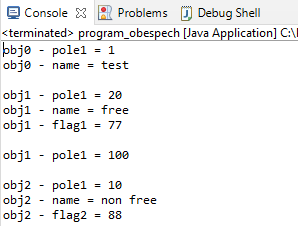


Рисунок 4