Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

**Лабораторная работа №5**

По дисциплине «СПП»

за пятый семестр

**Выполнил:**

Студент 3 курса

Факультета ЭИС

Группы ПО-3

Будяков В.В.

**Проверил:**

Крощенко А.А.

**Брест 2020 г.**

Вариант 3

Цель работы: приобрести практические навыки в области объектно-ориентированного проектирования.

Ход работы:

Реализовать абстрактные классы или интерфейсы, а также наследование и полиморфизм для следующих классов:

interface Сотрудник ← class Инженер ← class Руководитель.

Код алгоритма:

public interface employee

{

public void show();

public void payday();

}

public class engineer implements employee

{

String salaryDay;

public engineer(String \_salaryDay)

{

salaryDay = \_salaryDay;

}

public void show()

{

}

public void payday()

{

System.out.println(salaryDay);

}

}

public class director extends engineer

{

String name;

int age;

@Override

public void show()

{

System.out.println(name+" "+age+" years\n");

}

public director(String \_name, int \_age, String \_salaryDay)

{

super(\_salaryDay);

name = \_name;

age = \_age;

}

}

public class Lab5\_1

{

public static void main(String[] args)

{

director d = new director("Vasya", 25, "10 of October");

d.show();

d.payday();

}

}

Результат выполнения программы:



В следующих заданиях требуется создать суперкласс (абстрактный класс, интерфейс) и определить общие методы для данного класса. Создать подклассы, в которых добавить специфические свойства и методы. Часть методов переопределить. Создать массив объектов суперкласса и заполнить объектами подклассов. Объекты подклассов идентифицировать конструктором по имени или идентификационному номеру. Использовать объекты подклассов для моделирования реальных ситуаций и объектов.

Код алгоритма:

abstract class musicInst

{

private String instrument;

public musicInst(String \_inst)

{

instrument = \_inst;

}

public String getInst()

{

return instrument;

}

public abstract void show();

}

public class percussion extends musicInst

{

String owner;

public percussion(String \_inst, String \_owner)

{

super(\_inst);

owner = \_owner;

}

public void show()

{

System.out.println(owner+" plays on "+super.getInst());

}

}

public class strings extends musicInst

{

String owner;

public strings(String \_inst, String \_owner)

{

super(\_inst);

owner = \_owner;

}

public void show()

{

System.out.println(owner+" plays on "+super.getInst()+" for 2 years");

}

}

public class wind extends musicInst

{

String owner;

public wind(String \_inst, String \_owner)

{

super(\_inst);

owner = \_owner;

}

public void show()

{

System.out.println(owner+" plays on "+super.getInst()+" for 6 years");

}

}

import java.util.\*;

public class Lab5\_2

{

public static void main(String[] args)

{

Vector<musicInst> insts = new Vector<>();

insts.add(new percussion("drums", "Vasya"));

insts.add(new strings("guitar","Kolya"));

insts.add(new wind("trumpet", "Grysha"));

for (musicInst i: insts)

{

i.show();

}

}

}

Результат выполнения программы:



Вывод: приобрел практические навыки в области объектно-ориентированного проектирования.