Joao Paulo Dos Santos Ferreira

Fundamentals of Programing II – CSIT 112\_04

Professor Dajin Wang

April 2, 2018

**Programing Project 4/2/2018**

**PP 9.1 -** Write a class called *MonetaryCoin* that is derived from the *Coin* class presented in Chapter 5. Store an integer in the *MonetaryCoin* that represents its value and add a method that returns its value. Create a *main* driver class to instantiate and compute the sum of several *MonetaryCoin* objects. Demonstrate that a monetary coin inherits its parent’s ability to be flipped.

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Coin.java Author: Lewis/Loftus

//

// Represents a coin with two sides that can be flipped.

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**public** **class** Coin

{

**private** **final** **int** HEADS = 0;

**private** **final** **int** TAILS = 1;

**private** **int** face;

//-----------------------------------------------------------------

// Sets up the coin by flipping it initially.

//-----------------------------------------------------------------

**public** Coin ()

{

flip();

}

//-----------------------------------------------------------------

// Flips the coin by randomly choosing a face value.

//-----------------------------------------------------------------

**public** **void** flip ()

{

face = (**int**) (Math.*random*() \* 2);

}

//-----------------------------------------------------------------

// Returns true if the current face of the coin is heads.

//-----------------------------------------------------------------

**public** **boolean** isHeads ()

{

**return** (face == HEADS);

}

//-----------------------------------------------------------------

// Returns the current face of the coin as a string.

//-----------------------------------------------------------------

**public** String toString()

{

String faceName;

**if** (face == HEADS)

faceName = "Heads";

**else**

faceName = "Tails";

**return** faceName;

}

}

**public** **class** MonetaryCoin **extends** Coin {

// Creates a integer variable to store the coin value

**private** **int** coinValue;

// Constructor assigns coin value and calls the flip method inherited

// from the parent class

**public** MonetaryCoin (**int** coinValue)

{

**this**.coinValue = coinValue;

flip();

}

// returns the coin value

**public** **int** getvalue()

{

**return** coinValue;

}

}

**public** **class** CoinDriver {

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

// Creates MonetaryCoin objects and flips each coin

MonetaryCoin quarter = **new** MonetaryCoin(25);

MonetaryCoin dime = **new** MonetaryCoin(10);

MonetaryCoin penny = **new** MonetaryCoin(5);

// Calls the getValue method from the child class

System.***out***.println("One quarter + one dime + one penny = " + (quarter.getvalue() + dime.getvalue() + penny.getvalue()) + " cents");

// Prints the object, since there is no toString method defined in the child class

// Calls the toString method from the parent class

System.***out***.println("Fliping the quarter gives you: " + quarter);

System.***out***.println("Fliping the dime gives you: " + dime);

System.***out***.println("Fliping the penny gives you: " + penny);

}

}

**PP 9.2 -** Design and implement a set of classes that define the employees of a hospital: doctor, nurse, administrator, surgeon, receptionist, janitor, and so on. Include methods in each class that are named according to the services provided by that person and that print an appropriate message. Create a *main* driver class to instantiate and exercise several of the classes.

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// HospitalEmployee.java Authors: Lewis/Loftus

//

// Solution to Programming Project 9.2

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**public** **class** HospitalEmployee

{

**protected** String name;

**protected** **int** number;

//-----------------------------------------------------------------

// Sets up this hospital employee with the specified information.

//-----------------------------------------------------------------

**public** HospitalEmployee (String empName, **int** empNumber)

{

name = empName;

number = empNumber;

}

//-----------------------------------------------------------------

// Sets the name for this employee.

//-----------------------------------------------------------------

**public** **void** setName (String empName)

{

name = empName;

}

//-----------------------------------------------------------------

// Sets the employee number for this employee.

//-----------------------------------------------------------------

**public** **void** setNumber (**int** empNumber)

{

number = empNumber;

}

//-----------------------------------------------------------------

// Returns this employee's name.

//-----------------------------------------------------------------

**public** String getName()

{

**return** name;

}

//-----------------------------------------------------------------

// Returns this employee's number.

//-----------------------------------------------------------------

**public** **int** getNumber()

{

**return** number;

}

//-----------------------------------------------------------------

// Returns a description of this employee as a string.

//-----------------------------------------------------------------

**public** String toString()

{

**return** name + "\t" + number;

}

//-----------------------------------------------------------------

// Prints a message appropriate for this employee.

//-----------------------------------------------------------------

**public** **void** work()

{

System.***out***.println (name + " works for the hospital.");

}

}

**public** **class** Administrator **extends** HospitalEmployee {

// Child's constructor calls the Parent's constructor

**public** Administrator(String name, **int** number)

{

**super**(name, number);

}

// Method defines a service provided by an Administrator

**public** String overiview()

{

**return** "It is the administrator's job to overview all the activities in the hospital.";

}

}

**public** **class** Doctor **extends** HospitalEmployee {

// Child's constructor calls the Parent's constructor

**public** Doctor(String name, **int** number)

{

**super**(name, number);

}

// Method defines a service provided by a Doctor

**public** String diagnose()

{

**return** "It is the doctor's job to diagnose patients.";

}

}

**public** **class** Janitor **extends** HospitalEmployee {

// Child's constructor calls the Parent's constructor

**public** Janitor(String name, **int** number)

{

**super**(name, number);

}

// Method defines a service provided by a Janitor

**public** String cleaning()

{

**return** "It is the janitor's job to keep the hospital clean.";

}

}

**public** **class** Nurse **extends** HospitalEmployee {

// Child's constructor calls the Parent's constructor

**public** Nurse(String name, **int** number)

{

**super**(name, number);

}

// Method defines a service provided by a Nurse

**public** String medAdmin ()

{

**return** "It is the nurse's job to administer medication.";

}

}

**public** **class** Receptionist **extends** HospitalEmployee {

// Child's constructor calls the Parent's constructor

**public** Receptionist(String name, **int** number)

{

**super**(name, number);

}

// Method defines a service provided by a Receptionist

**public** String answeringPhone ()

{

**return** "It is the receptionist job to answer the phone.";

}

}

**public** **class** Surgeon **extends** HospitalEmployee {

// Child's constructor calls the Parent's constructor

**public** Surgeon(String name, **int** number)

{

**super**(name, number);

}

// Method defines a service provided by a Surgeon

**public** String surgery()

{

**return** "It is the surgeon's job to do surgeries.";

}

}

// Driver class

**public** **class** HospitalDriver {

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

// Creates one of each object from the child classes

Administrator admin1 = **new** Administrator("Allison", 1001);

Doctor doc1 = **new** Doctor("Robert", 2001);

Janitor jan1 = **new** Janitor("Carl", 3001);

Nurse nur1 = **new** Nurse("Will", 4001);

Receptionist recep1 = **new** Receptionist("Joyce", 5001);

Surgeon surg1 = **new** Surgeon("Paul", 6001);

// Calls a method from the parent class using the child object and a unique method from each child class

admin1.work();

System.***out***.println("She's an administrator. " + admin1.overiview() + "\n");

doc1.work();

System.***out***.println("He's a Doctor. " + doc1.diagnose() + "\n");

jan1.work();

System.***out***.println("He's a Janitor. " + jan1.cleaning() + "\n");

nur1.work();

System.***out***.println("He's a Nurse. " + nur1.medAdmin() + "\n");

recep1.work();

System.***out***.println("She's a Receptionist. " + recep1.answeringPhone() + "\n");

surg1.work();

System.***out***.println("He's a Surgeon. " + surg1.surgery() + "\n");

}

}