(Лаборатори №1)

Н. Мөнхжин

ХШУИС, Компьютерийн ухааны оюутан, nyamkamunhjin@gmail.com

1. ОРШИЛ/УДИРТГАЛ

Програмчлалын хэлний ойлголт болох объект хандлагат технологийн класс, объект, битүүмжлэл, классын гишүүн өгөгдөл, гишүүн функцийн хүрээнд даалгаруудыг хийж гүйцэтгэв.

2. ЗОРИЛГО

Объект хандлагат програмчлалыг хүрээнд дараах зорилтуудыг тавьж ажилласан.

1. Объектийг загварлах класс үүсгэх.
2. Гишүүн өгөгдөл, гишүүн функцуудыг зарлах, тодорхойлох.
3. Гишүүн өгөгдөл, функцуудыг битүүмжлэх.

3. ОНОЛЫН СУДАЛГАА

3.1 Класс ба объект

Объектыг програмын орчинд хийсвэрлэхдээ Класс гэсэн ойлголтыг ашигладаг . Объект гэдэг нь хүн, амьтан, машин, банкны данс гэх зэрэг өөрийн гэсэн шинжтэй, үйлдэлтэй зүйлсүүдийг хэлнэ. [1]

Объектын шинжийг классын гишүүн өгөгдлөөр төлөөлүүлдэг. Харин хийдэг үйлдлүүдийг гишүүн функцээр төлөөлүүлдэг. Жишээ нь: Тухайн хүний өндөр 180, жин 70кг, унтдаг, иддэг г.м.

Битүүмжлэл нь объектын өгөгдлийг бусдаас далдалж байдаг. Тухайн объект яаж загварчлагдсанаас хамааран гишүүн өгөгдлийг өөр объект мэдэх шаардлагагүй байдаг. Энэ нь программын найдвартай байдлыг хангадаг.[2]

4. ХЭРЭГЖҮҮЛЭЛТ

1. Класс гэж юу болох, онцлог, үүргийн талаар бич.

- Класс нь объектыг программын орчинд загварчлахад ашигладаг ойлголт юм.

- Объектын шинж тэмдэг, үйл хөдлөлийг класс агуулна.

2. Класс болон объектын ялгаа?

- Объект нь өөрийн гэсэн шинж тэмдэгтэй, өвөрмөц үйлдэлтэй байдаг бол Класс нь объектыг загварчлах хийсвэр ойлголт юм.

3. Гишүүн функц, гишүүн өгөгдөл хоёр ямар хамааралтай байдаг вэ? Өгөгднийн битүүмжлэл гэж юу вэ?

- Объектын гишүүн функц нь тухайн объектын өгөгдөл дээр хийгддэг үйлдлүүдийг хэлнэ. Жишээ нь: Ажилчин классын ажилчны цалин харуулах функц болно. Энэ функц нь гишүүн өгөгдөл “цалин”-д хандаж утга буцаана.

4. Классын гишүүн өгөгдөл болон гишүүн функцэд хэрхэн хандах вэ?

- Хэрвээ өгөгдлийг битүүмжлээгүй бол үүсгэсэн объектын ард цэг тавин өгөгдлийн нэрийг бичнэ. Битүүмжилсэн бол тухайн өгөгдөлд getters, setters буюу утга оноох, утга авах public функцуудыг класс дотор зарлаж өгөх хэрэгтэй.
- Java хэл дээр класс дотор өгөгдөлтэй хандахын тулд түлхүүрийг “this” ашиглана. (this.name) гэх мэт.[3]
- Гишүүн функц-д гаднаас хандахын тулд заавал public битүүмжлэлтэй байх ёстой.

5. Ажилчин гэсэн класс тодорхойлно. Ажилчдын ажилласан цаг бүрийг өөрчилж цалинг тооцоолох жижиг програм бич.

Ажилчин классыг үүсгэх.

```
Class Employee {

    // дотор нь гишүүн өгөгдөл буюу шинжүүдийг тодорхойлох.

    Private int id;

    private String name;

    private String occupation;

    private float workedHours;

    // гишүүн функцуудыг тодорхойлох.!

    Public void inputInfo();

    public void showInfo();

    public void calculateSalary();

    public void calculateBossSalary();

    public void addHours(float hours); гэх мэт үүсгэж доторх логикуудыг
    бичнэ.

}
```

5. ДҮГНЭЛТ

Объект болон Классыг программ бичихдээ ашигласнаар код илүү цэгцтэй ойлгомжтой болж, асуудлыг шийдвэрлэхэд илүү хялбар, дахин ашиглахад амархан болж байна.

6. АШИГЛАСАН МАТЕРИАЛ

1. Объект хандлагат технологийн С++ програмчлал, Ж.Пүрэв, 2008, Улаанбаатар. Хуудас 3.
2. Объект хандлагат технологийн С++ програмчлал, Ж.Пүрэв, 2008, Улаанбаатар. Хуудас 4
3. https://docs.oracle.com/javase/tutorial/java/javaOO/thiskey.html

7. ХАВСРАЛТ

```
import java.util.Scanner;

public class Main {

    public static void main(String[] args) {

      // write your code here

        Employee sam = new Employee(123, "sam", "manager", 34);

        sam.inputInfo();

        sam.showInfo();

    }

}


class Employee {

    // fields

    private int id;

    private String name;

    private String occupation;

    private float workedHours;


    Employee(int id, String name, String occupation, float workedHours) {

        this.id = id;

        this.name = name;

        this.occupation = occupation;

        this.workedHours = workedHours;

    }
```

```
//salaries for employees

private final float SALARY_BOSS = 30;
private final float SALARY_MANAGER = 17;
private final float SALARY_EMPLOYEE = 10;

public void inputInfo() {

    System.out.print("Enter id: ");
    this.id = (new Scanner(System.in)).nextInt();

    System.out.print("\nEnter name: ");
    this.name = (new Scanner(System.in)).next();

    System.out.print("\nEnter occupation: ");
    this.occupation = (new Scanner(System.in)).next();

    System.out.print("\nEnter workedHours: ");
    this.workedHours = (new Scanner(System.in)).nextFloat();
}

public void showInfo() {
    System.out.println("id:" + this.id);
    System.out.println("name: " + this.name);
    System.out.println("occupation:" + this.occupation);
```

```
        System.out.println("worked hours: " + this.workedHours);
    }


    public float calculateSalary() {
        float salary = 0;
        switch(this.occupation.toLowerCase()) {
            case "boss":
                salary =  bossSalaryCalculate();
                break;


            case "manager":
                salary = this.workedHours * SALARY_MANAGER;
                break;


            case "employee":
                salary =  this.workedHours * SALARY_EMPLOYEE;
                break;
        }


        if(salary == 0)
            System.out.println("Does he work here?");


        return salary;
    }
```

```
    public float bossSalaryCalculate() {
        if(this.occupation.toLowerCase().equals("boss"))
            return this.workedHours * SALARY_BOSS;
        else {
            System.out.println("You are not boss!");
            return 0;
        }

    }

    public boolean addWorkHours(float hours) {
        if(this.occupation.toLowerCase().equals("employee")
            || this.occupation.toLowerCase().equals("boss")
            || this.occupation.toLowerCase().equals("manager")) {

            this.workedHours += hours;

            return true;
        }
        else
            return false;
    }
}
```