Programlama Dillerinin Prensipleri

Lab Notları – 10

Nesne Yönelimli Programlama -2

Java'da Soyut Sınıflar

```
package Mat;

import java.util.Date;

/**
 *
 * GeometrikSekil.Java
 */
public abstract class GeometrikSekil {
   private String renk = "Mavi";
   private Date olusturulmaTarihi;

   protected GeometrikSekil(){
      olusturulmaTarihi = new Date();
   }
   public void setRenk(String renk){
      this.renk = renk;
   }
   public Date OlusturulmaTarihi(){
      return olusturulmaTarihi;
   }
   @Override
   public String toString() {
      String cikti = "Tarih: "+olusturulmaTarihi;
      cikti += "\nRenk: "+renk;
      cikti += "\nAlan: "+Alan();
      cikti += "\nCevre: "+Cevre();
      return cikti;
   }
   public abstract double Alan();
   public abstract double Cevre();
}
```

```
package Mat;

/**
 * Daire.Java
 */
public class Daire extends GeometrikSekil{
   private double yaricap;
   public Daire(double yaricap){
      this.yaricap = yaricap;
   }
```

```
    @Override
    public double Cevre() {
        return 2*Math.PI*yaricap;
    }

    @Override
    public double Alan() {
        return Math.PI*Math.pow(yaricap, 2);
    }

}
```

```
package Mat;

/**
 * Kare.Java
 */
public class Kare extends GeometrikSekil {
    private double kenar;
    public Kare(double kenar){
        this.kenar = kenar;
    }

    @Override
    public double Cevre() {
        return 4*kenar;
    }

    @Override
    public double Alan() {
        return Math.pow(kenar, 2);
    }
}
```

```
package proje2;

import Mat.*;
public class Proje2 {
    public static void main(String[] args) {
        Daire d = new Daire(12);
        System.out.println(d);
        Kare k = new Kare(5);
        System.out.println(k);
    }
}
```

C Dilinde Soyut Sınıfların Benzetilmesi

```
// GeometrikSekil.h dosyası
#ifndef GEOMETRIKSEKIL_H
#define GEOMETRIKSEKIL_H

#include "stdio.h"
#include "stdlib.h"
#include "time.h"

struct GEOMETRIKSEKIL{
	char* renk;
	char* olusturulmaTarihi;

	void (*SetRenk)(struct GEOMETRIKSEKIL*,char*);
	char* (*Tarih)(struct GEOMETRIKSEKIL*);
	void (*Yaz)(struct GEOMETRIKSEKIL*,void*);
	double (*Alan)();
	double (*Cevre)();
	void (*Yoket)(struct GEOMETRIKSEKIL*);
};
typedef struct GEOMETRIKSEKIL* GeometrikSekil;

GeometrikSekil GeometrikSekilOlustur();
void setRenk(const GeometrikSekil,char*);
char* OlusturulmaTarihi(const GeometrikSekil);
void SekilYaz(const GeometrikSekil,void*);
void GeometrikSekilYoket(GeometrikSekil);

#endif
```

```
// GeometrikSekil.c dosyası
#include "GeometrikSekil.h"

char* ZamanGetir(){
	char* buff = malloc(sizeof(char)*100);
	time_t simdi = time(0);
	strftime(buff,100,"%Y-%m-%d %H:%M",localtime(&simdi));
	return buff;
}

GeometrikSekil GeometrikSekilOlustur(){
	GeometrikSekil this;
	this = (GeometrikSekil)malloc(sizeof(struct GEOMETRIKSEKIL));
	this->renk = "Mavi";
	this->olusturulmaTarihi = ZamanGetir();
	this->SetRenk = &setRenk;
	this->Tarih = &OlusturulmaTarihi;
	this->Yaz = &SekilYaz;
	this->Yoket = &GeometrikSekilYoket;
	return this;
```

```
}
void setRenk(const GeometrikSekil this,char* renk){
	this->renk = renk;
}
char* OlusturulmaTarihi(const GeometrikSekil this){
	return this->olusturulmaTarihi;
}
void SekilYaz(const GeometrikSekil this,void *p){
	printf("\n\nTarih: %s",this->olusturulmaTarihi);
	printf("\nRenk: %s",this->renk);
	printf("\nAlan: %lf",this->Alan(p));
	printf("\nCevre: %lf",this->Cevre(p));
}
void GeometrikSekilYoket(GeometrikSekil this){
	if(this == NULL) return;
	free(this->olusturulmaTarihi);
	free(this);
}
```

```
#ifndef DAIRE_H
#define DAIRE_H

#include "GeometrikSekil.h"
#include "Math.h"

struct DAIRE{
	GeometrikSekil super;
	double yaricap;
	void (*Yoket)(struct DAIRE*);
};
typedef struct DAIRE* Daire;

Daire DaireOlustur(double);
double Alan(const Daire);
double Cevre(const Daire);
void DaireYoket(Daire);

#endif
```

```
#include "Daire.h"


Daire DaireOlustur(double yaricap){
	Daire this;
	this = (Daire)malloc(sizeof(struct DAIRE));
	this->super = GeometrikSekilOlustur();
	this->yaricap = yaricap;
	this->super->Alan = &Alan;
	this->super->Cevre = &Cevre;
	this->Yoket = &DaireYoket;
	return this;
}
```

```
double Alan(const Daire this){
	return M_PI*pow(this->yaricap,2);
}
double Cevre(const Daire this){
	return 2*M_PI*this->yaricap;
}
void DaireYoket(Daire this){
	if(this == NULL) return;
	this->super->Yoket(this->super);
	free(this);
}
```

```
#ifndef KARE_H
#define KARE_H

#include "GeometrikSekil.h"
#include "Math.h"

struct KARE{
	GeometrikSekil super;
	double kenar;
	void (*Yoket)(struct KARE*);
};
typedef struct KARE* Kare;

Kare KareOlustur(double);
double alan(const Kare);
double cevre(const Kare);
void KareYoket(Kare);
#endif
```

```
#include "Kare.h"

Kare KareOlustur(double kenar){
	Kare this;
	this = (Kare)malloc(sizeof(struct KARE));
	this->super = GeometrikSekilOlustur();
	this->kenar = kenar;
	this->super->Alan = &alan;
	this->super->Cevre = &cevre;
	this->Yoket = &KareYoket;
	return this;
}
double alan(const Kare this){
	return pow(this->kenar,2);
}
double cevre(const Kare this){
	return 4*this->kenar;
}
void KareYoket(Kare this){
	if(this == NULL) return;
	this->super->Yoket(this->super);
```

```
        free(this);
}
```

```
#include "Daire.h"
#include "Kare.h"

int main(){
        Daire daire = DaireOlustur(4.8);
        daire->super->Yaz(daire->super,daire);

        Kare kare = KareOlustur(10);
        kare->super->Yaz(kare->super,kare);

        daire->Yoket(daire);
        kare->Yoket(kare);

        return 0;
}
```

```
hepsi: derle calistir

derle:
        gcc -I ./include/ -o ./lib/GeometrikSekil.o -c ./src/GeometrikSekil.c
        gcc -I ./include/ -o ./lib/Daire.o -c ./src/Daire.c
        gcc -I ./include/ -o ./lib/Kare.o -c ./src/Kare.c
        gcc -I ./include/ -o ./bin/Test ./lib/GeometrikSekil.o ./lib/Daire.o ./lib/Kare.o ./src/Test.c

calistir:
        ./bin/Test
```

Java'da Arayüzler

Arayüz sadece sabitlerin ve soyut metotların tanımlandığı sınıf benzeri yapılardır. Soyut sınıflara benzerler fakat soyut sınıflar soyut olmayan metotlarda (gövdesi olan metotlar) içerdiği için bu yönüyle arayüzlerden ayrılmaktadırlar. Arayüzlerden nesne oluşturulamaz.

Java'da arayüzler özel bir sınıf gibi davranırlar ve her arayüz derlenme sonrasında bytecode dosyası oluşur. Arayüzde tanımlanan alanlara public dışında niteleyici getirilemez.

```
package Paket;
public interface Sekil {
   double Alan();

}
```

```
package Paket;

public class Dikdortgen implements Sekil{
   private int genislik;
   private int yukseklik;

   public Dikdortgen(int genislik, int yukseklik){
      this.genislik = genislik;
```

```
        this.yukseklik = yukseklik;
    }

    @Override
    public double Alan(){
        return genislik*yukseklik;
    }
}
```

```
package Paket;

public class Daire implements Sekil {
    private double yaricap;
    public Daire(double yaricap){
        this.yaricap = yaricap;
    }
    @Override
    public double Alan(){
        return Math.PI * Math.pow(yaricap, 2);
    }
}
```

```
import Paket.*;

public class Arayuz {

    public static void main(String[] args) {
        // TODO code application logic here
        Dikdortgen d = new Dikdortgen(25, 3);
        System.out.println(d.Alan());

        Daire da = new Daire(3);
        System.out.println(da.Alan());
    }

}
```

Java'daki Object Türünün C Dilinde Benzetilmesi

```
#include "Daire.h"
#include "Kare.h"

typedef void* Object;

int main(){
        Object nesne = DaireOlustur(5);
        printf("Daire'nin Cevre ve Alani\n");
        printf("Daire Cevre: %.2lf\n",((Daire)nesne)->super->Cevre((Daire)nesne));
        printf("Daire Alan:  %.2lf\n",((Daire)nesne)->super->Alan((Daire)nesne));
        ((Daire)nesne)->Yoket((Daire)nesne);

        nesne = KareOlustur(10);
        printf("Kare'nin Cevre ve Alani\n");
```

```
        printf("Kare Cevre: %.2lf\n",((Kare)nesne)->super->Cevre((Kare)nesne));
        printf("Kare Alan:  %.2lf\n",((Kare)nesne)->super->Alan((Kare)nesne));

        ((Kare)nesne)->Yoket((Kare)nesne);
        return 0;
}
```

Hazırlayan
Dr. Öğr. Üyesi M. Fatih ADAK