С. Торайгыров атындағы Павлодар мемлекеттік университеті

Java объектілері мен негізгі кластар

Кластар

- Java кластар тілінде біріктіру класс өрістері, әдістері, конструкторлар, логикалық блоктар және ішкі кластар.
- C++ кластың негізгі айырмашылықтары:
- барлық функциялар класс ішінде анықталады және әдістер деп аталады;
- Әдісті құру мүмкін емес, класс әдісі болып табылмайтын, және кластан тыс әдісін жариялау;
- Спецификаторы доступа public, private, protected воздействуют только на те объявления полей, методов и классов, перед которыми они стоят, а не на участок от одного до другого спецификатора, как в C++;
- private элементтері бойынша әдепкі белгіленбейді, қол жетімді осы класс топтамасы үшін. Хабарландыру класының түрі бар:

```
[спецификаторы] class ИмяКласса [extends СуперКласс] [implements список_интерфейсов] {
        /* определение класса */
}
```

Кластар

- Спецификатор классқа қол жеткімділігі мүмкін public (класс қол жетімді пакетте және топтамадан тыс), final (класстын подклассы болуы мүмкін емес), abstract (класс абстрактілі әдістерін қамтуы мүмкін, мұндай дәрежеде класс құруға болмайды).
- Әдетте класс спецификаторі қойылмаған болса, ол достықпен орнатылады (friendly).
- Мұндай класс тек ағымдағы пакетте қол жетімді.
- friendly спецификаторы жариялау кезінде мүлдем пайдаланылмайды және негізгі түйінді сөз тілі емес.
- Бұл сөз бағдарламашылар сленгесінде пайдаланылады, яғни әдеттегі мәнін қысқаша белгілеу үшін.

- Класс барлық қасиеттер мен әдістерді суперкласстан мұра етеді көрсетілген **extends** түйінді сөзінен кейін, және көптеген интерфейстерді қамтуы мүмкін. Implements түйінді сөзінен кейін үтір арқылы.
- Интерфейстер абстрактілі кластарға өте ұқсас, тек қана тұрақтылар және сигнатура әдістері іске асыру мүмкіндігінсіз.

Кластар

- Барлық кластар кез-келген қосымшаның шартты түрде екі топқа бөлінеді: **кластар — ақпарат тасымалдаушылар мен кластар, жұмыс істейтін ақпараттармен.**
- Кластар ақпаратты меңгерген пәндік облыс туралы деректер қосымшасын қамтиды.
- Мысалы, қосымша әуе қозғалысын басқару үшін болса, онда пәндік облысы ұшақтар мен жолаушылар болады, және т.б.
- Жобалау кезінде ақпараттық сарапшылардың классы маңызды инкапсуляция,класс алаңының нақты ақпаратын қамтамасыз ететін мәндер.
- Инкапсуляция көмегімен негізгі private сөзі тікелей қол жеткізу мәні объект арқылы жабылады, өріс мағынасының инициализациясына жауапты әдіс, кіріс мағынаны нақты ету үшін орындайтын тексеріс.

Мысал ретінде бұза отырып, инкапсуляция қарастыруға болады Coin қосымшалары бойынша монеталарды өңдеу.

```
package by.bsu.fund.bean;
public class Coin {
        public double diameter; // нарушение инкапсуляции
        private double weight; // правильная инкапсуляция
        public double getDiameter() {
                return diameter;
        }
        public void setDiameter(double value) {
                if (value > 0) {
                        diameter = value;
                } else {
                        diameter = 0.01; // значение по умолчанию
                }
        }
        public double takeWeight() { // некорректно: неправильное имя метода
                return weight;
        }
        public void setWeight(double value) {
                weight = value;
        }
}
```

Coin классы

- **Coin** класы құрамында екі өріс **diameter және weight**, **public және private** ретінде белгіленген.
- Өрістің мағынасы **weight** әдісі арқылы өзгертуге болады, мысалы, **setWeight (double value).**
- Кейбір жағдайларда ауыстырудың әдіпсіз маңызы бар мән-әдепкі әкелуі мүмкін, алдағы уақытта өріскен қателіктерге душар болуы мүмкін, сондықтан генерация алып тастаудың орнына жиі ауыстыру жүргізіледі.
- Өріс **diameter** арқылы қолжетімді **Coin** класының тікелей объектісі. Өріс, осы тәсілмен жарияланған, деп саналатын «**тугой**» инкапсуляциясын бұза отырып жарияланады, нәтижесі бұзылуы мүмкін ақпарат ретінде төменде көрсетілген:

```
package by.bsu.fund.run;
import by.bsu.fund.bean.Coin;
public class Runner {
        public static void main(String[ ] args) {
                Coin ob = new Coin();
                ob.diameter = -0.12; // некорректно: прямой доступ
                ob.setWeight(100);
               // ob.weight = -150; // поле недоступно: compile error
        }
}
```

- ob.diameter = -0.12; мүмкін емес diameter өрісі Coin классына private double diameter түрінде жариялау;
- онда жол берумен тікелей меншіктеу маңызы бар өрістің көмегімен объект сілтемелері компилиция қателігіне әкеледі.

```
package by.bsu.fund.bean;
public class Coin {
        private double diameter; // правильная инкапсуляция
        private double weight; // правильная инкапсуляция
        public double getDiameter() {
                return diameter;
        }
        public void setDiameter(double value) {
                if(value > 0) {
                   diameter = value;
                } else {
                   System.out.println("Отрицательный диаметр!");
                }
        }
        public double getWeight() { // правильное имя метода
                return weight;
        }
        public void setWeight(double value) {
                weight = value;
        }
}
```

- Дұрыстығын тексеру сырттан кіретін ақпарат жүзеге асырылатын әдісі setDiameter(double value) және объектінің баптандыру мүмкіндік беретін бұзылуы туралы хабардар етуге міндетті. Қол жеткізу public-әдістері кластың объектісі ғана жүзеге асырылатын кейін осы кластың объектісін құру.

объектіні құру, қолжетімділік өрістер мен объектінің әдістері# CompareCoin.java # Runner.java */

```
package by.bsu.fund.action;
import by.bsu.fund.bean.Coin;
public class CompareCoin {
        public void compareDiameter(Coin first, Coin second) {
                double delta = first.getDiameter() - second.getDiameter();
                if (delta > 0) {
                        System.out.println("Первая монета больше второй на " + delta);
                } else if (delta == 0) {


                        System.out.println("Монеты имеют одинаковый диаметр");
                 }  else {
                        System.out.println("Вторая монета больше первой на " + -delta);
                 }
        }
}
package by.bsu.fund.run;
import by.bsu.fund.bean.Coin;
import by.bsu.fund.action.CompareCoin;
public class Runner {
         public static void main(String[ ] args) {
                 Coin ob1 = new Coin();
                 ob1.setDiameter(-0.11); // сообщение о неправильных данных
                 ob1.setDiameter(0.12); // корректно
                 ob1.setWeight(150);
                 Coin ob2 = new Coin();
                 ob2.setDiameter(0.21);
                 ob2.setWeight(170);
                 CompareCoin ca = new CompareCoin();
                 ca.compareDiameter(ob1, ob2);
         }
}
```

- Компиляция и выполнение данного кода приведут к выводу на консоль следующей информации:
- *Теріс диаметрі! Вторая монета больше первой на 0.09.*
- Объектінің құрылуы екі қадамда құрылады.
- Алдымен класс сілтеме объектісі жарияланады. Содан кейін оператордың көмегімен new объектінің данасымен құрылады, мысалы:
- **Coin ob1;** // хабарландыру сілтемелері
- **ob1 = new Coin();** // объекті құру
- Дегенмен, осы екі әрекет бір әдетте біріктіреді: **Coin ob1 = new Coin();** /* хабарландыру сілтемелері мен объектіні құру */

- Оператор **new** конструкторын тудырады, ал мысалда конструктор әдепкі параметрсіз, бірақ жақшаның ішінде орналасуы мүмкін дәлелдер берілетін конструкторлар, онда класс параметрлерінің құрастырылуымен жарияланады.
- Операция объектілерді меншіктеу үшін білдіреді, бұл екі сілтеме бір учаскіні еске алудың көрсетуі болады.
- ***compareDiameter(Coin first, Coin second)*** *әдісі*
- Екі әдіс орындалады, олар осылай бөлінуі керек: салыстыра орындайды және есепті оқиды.
- Әрекет тым әр-түрлі табиғи болып табылуы бойынша, аралас болуы мүмкін.

- Салыстыру әдісі дана бір өріс# */
- public int compareDiameter(Coin first, Coin second) { int result = 0; double delta = first.getDiameter() - second.getDiameter(); if (delta > 0) { result = 1; } else if (delta < 0) { result = -1; } return result; }

- Есепті қалыптастыру керек басқа әдісті басқа класқа орналастыру үшін.