

АБСТРАКЦИЯ

ЖӘНЕ МҰРАЛАУ



Show More



dauclocloudlab@gmail.com

Get Started →

Дәрісіміздің мақсаты



Абстракция



Абстракция ұғымын
түсіндіреміз



Мұралау



Мұралау принципімен
танысамыз



Инкапсуляция

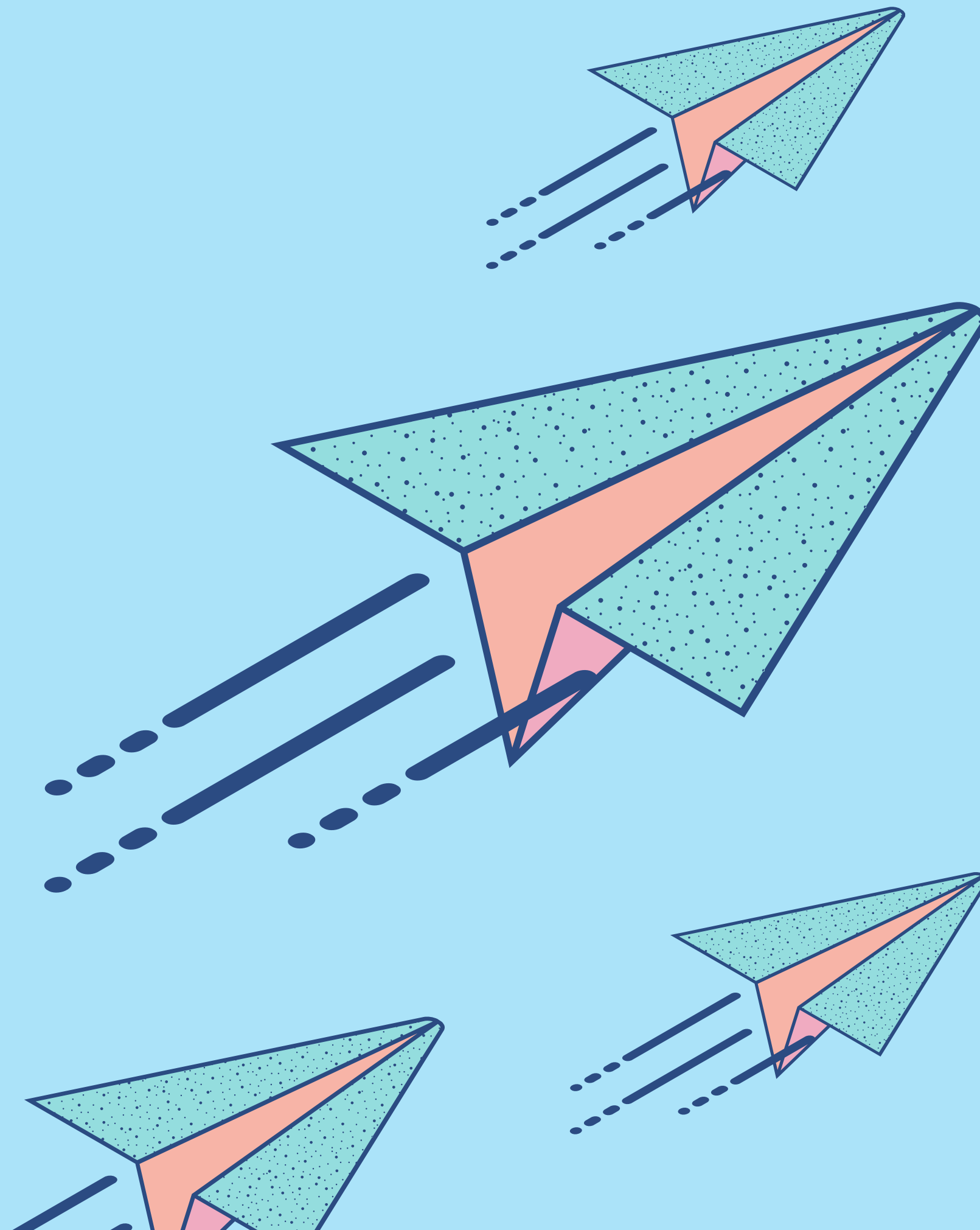


Абстракция мен
инкапсуляция ұғымдарын
салыстырамыз

Next Page →

Бастаймыз

Абстракция концепциясы



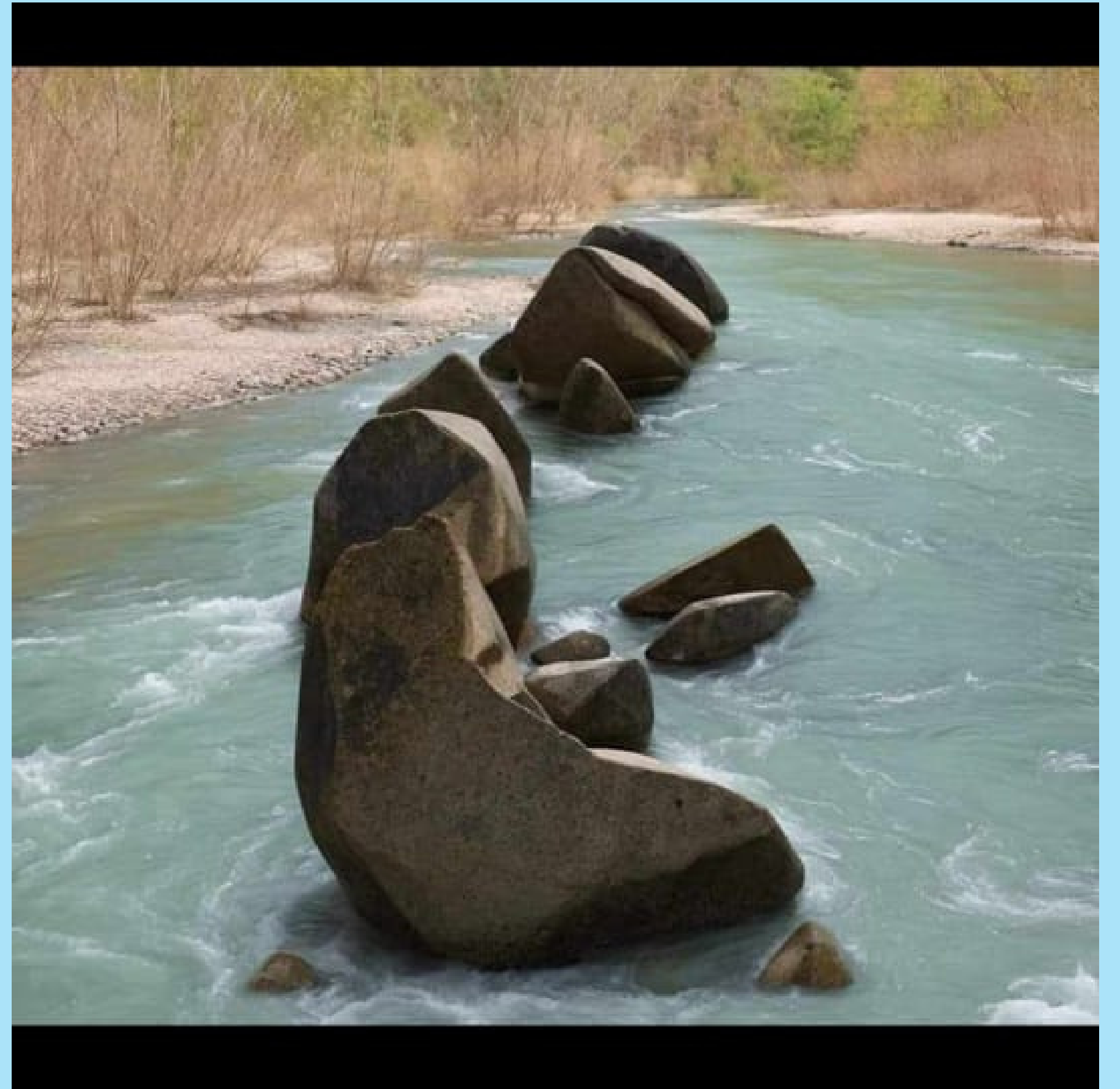


АБСТРАКЦИЯ

Инкапсуляция ұғымымен тығыз байланысы бар объектіге бағытталған программалаудың тағы бір принципі бұл - абстракция ұғымы.

Инкапсуляция ұғымы сияқты абстракция ұғымы да бірнеше мағыналы ұғым.

Сонымен қатар, абстракция ұғымы мұралау принципімен де тығыз байланысты.



Абстракция-1

Нақты объектінің маңызды деген қасиеттер жиынтығын қалдырып, маңызды емес деген қасиеттерді елемей қарапайым баламасын жасау.

Объектінің шынайы моделін сипаттап жазу тым күрделі, тіпті мүмкін емес тірлік. ОБП парадигмасында біздер қарапайым компромиске барамыз және өз жұмысымызды жеңілдетеміз. Егер объектілердің қасиеттері бірдей болса, ол объектілер бір классқа жатады деп санаймыз және керісінше бір класс объектілерінің қасиеттері бірдей болады деп айтамыз.

Мысалы, телефонды класс қасиеттерінің шекті жиынтығымен сипаттау, нақты өмірдегі телефонның қарапайым баламасы болып табылады.

Маңызды деген қасиеттер жиынтығын қалдырып, маңызды емес деген қасиеттерді елемей қарапайым баламасын жасауға біз әдейі барамыз. Бұл процессті **абстракциялау** деп атаймыз.

Нәтижесінде қарапайым, жеңілдетілген, идеализацияланған, абстракцияланған объект пайда болады. Мұндай абстракцияланған объектілерде маңызды деп тапқан қасиеттерді қалдырамыз және бұл маңызды деп тапқан қасиеттер шешуі тиіс мәселеге байланысты олардың жиынтығы әр түрлі болады.

Абстракция-1

бұл шың өмірден алынған бір объектінің немесе құбылыстың кейбір маңызды емес бөліктерін алмай немесе дәл осы контексте маңызы жоқ бөліктерін қалдыра құрылған модель.

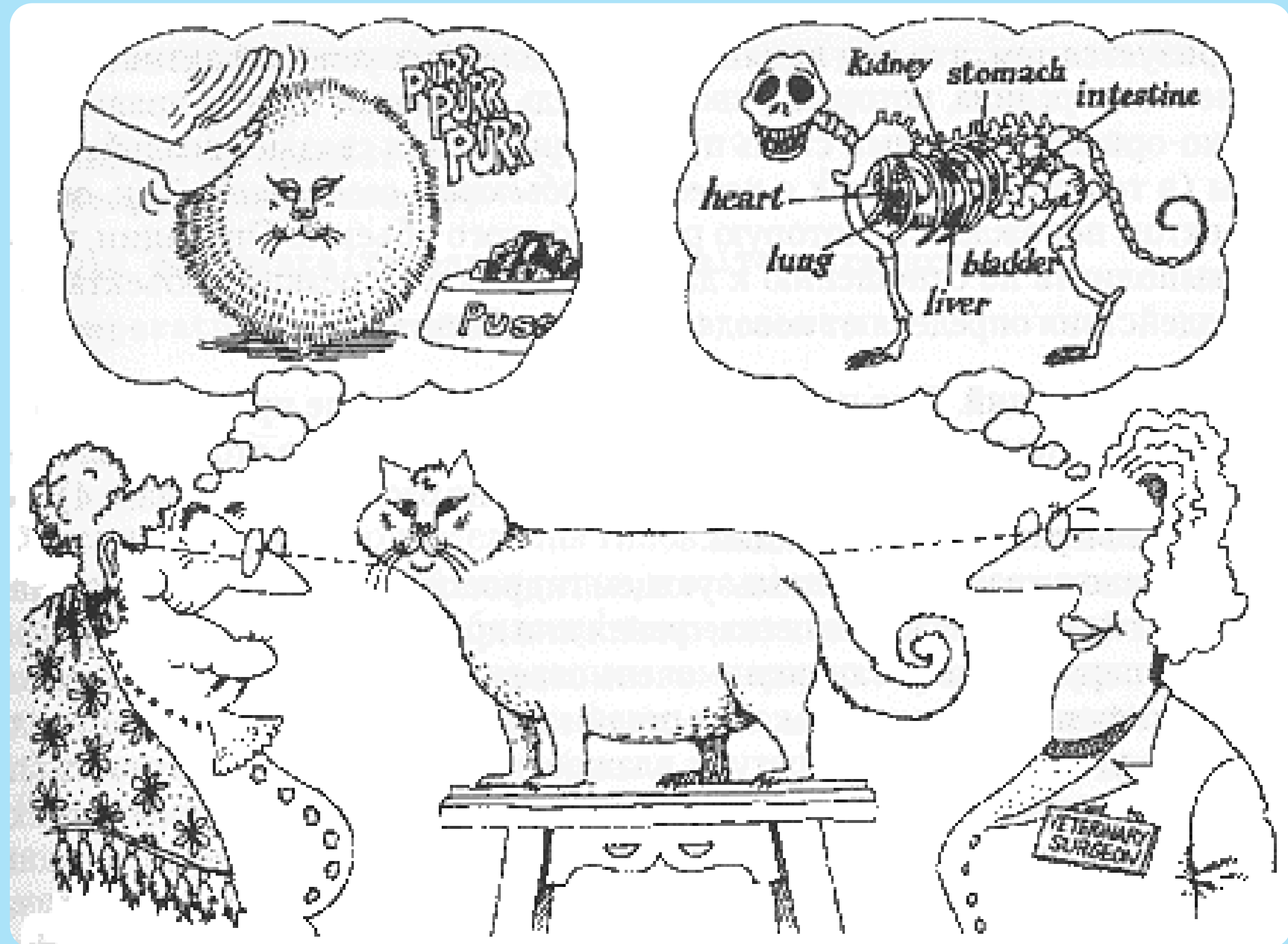
Объектінің абстракциялау дәрежесі де әр түрлі деңгейде болады: объектінің қасиеттер жиынтығы қаншалықты аз қалады, соншалықты болуы мүмкін қасиеттерді есепке алмаймыз деген сөз, демек абстракциялау дәрежесі соншалықты жоғары деген сөз. Объектілердің болуы мүмкін әртүрлілігінен абстракцияланамыз.

Ұқсас объектілерді жиын ретінде әр түрлі класстарға топтастыратын болсақ, біздер нақтылықтан (шыңайлықтан) әр түрлі деңгейде абстракцияланған класстарды алатын боламыз. Мысалы, әр түрлі кітаптарды жалпы “кітап” деп аталатын классқа жатқызуға да болады немесе “жоғарғы математика оқулығы”, “тарих оқулығы”, “әдебиет оқулы” деген сияқты класстарға топтастыруға болады.

Абстракция-1

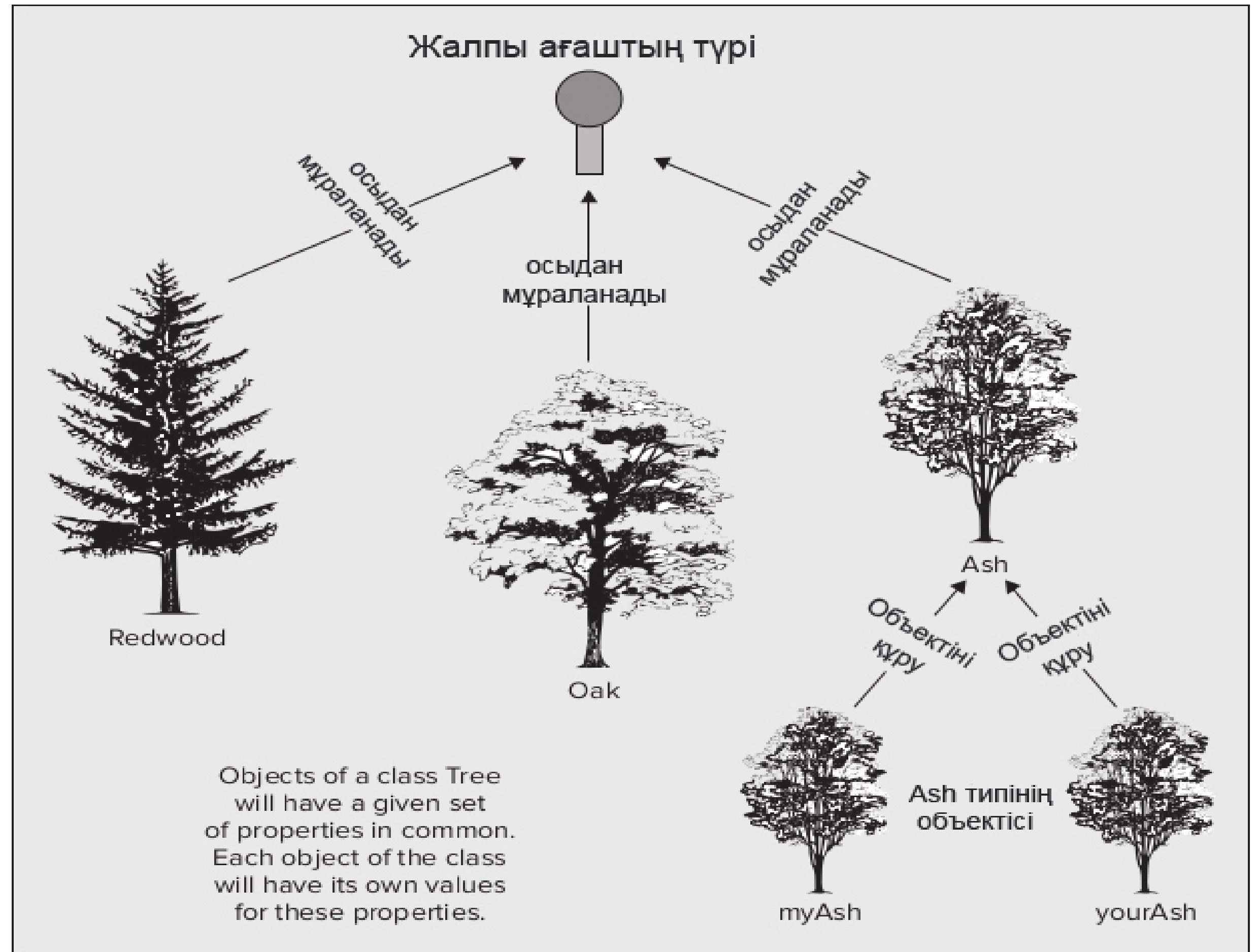
Нақты объектінің маңызды деген қасиеттер жиынтығын қалдырып, маңызды емес деген қасиеттерді елемей қарапайым баламасын жасау.

Бұл шың өмірден алынған бір объектінің немесе құбылыстың кейбір маңызды емес бөліктерін алмай немесе дәл осы контекісте маңызы жоқ бөліктерін қалдыра құрылған модель.



Мұралану

Нақты өмірдегі барлық объектілерді класс бойынша жіктейтін болсақ, онда класстар да әр түрлі болып шығатын еді. Сонда да кейбір класстар басқа класстармен салыстырғанда бір-біріне ұқсасырақ болар еді, себебі оларда ортақ қасиеттер көбірек болуы мүмкін.



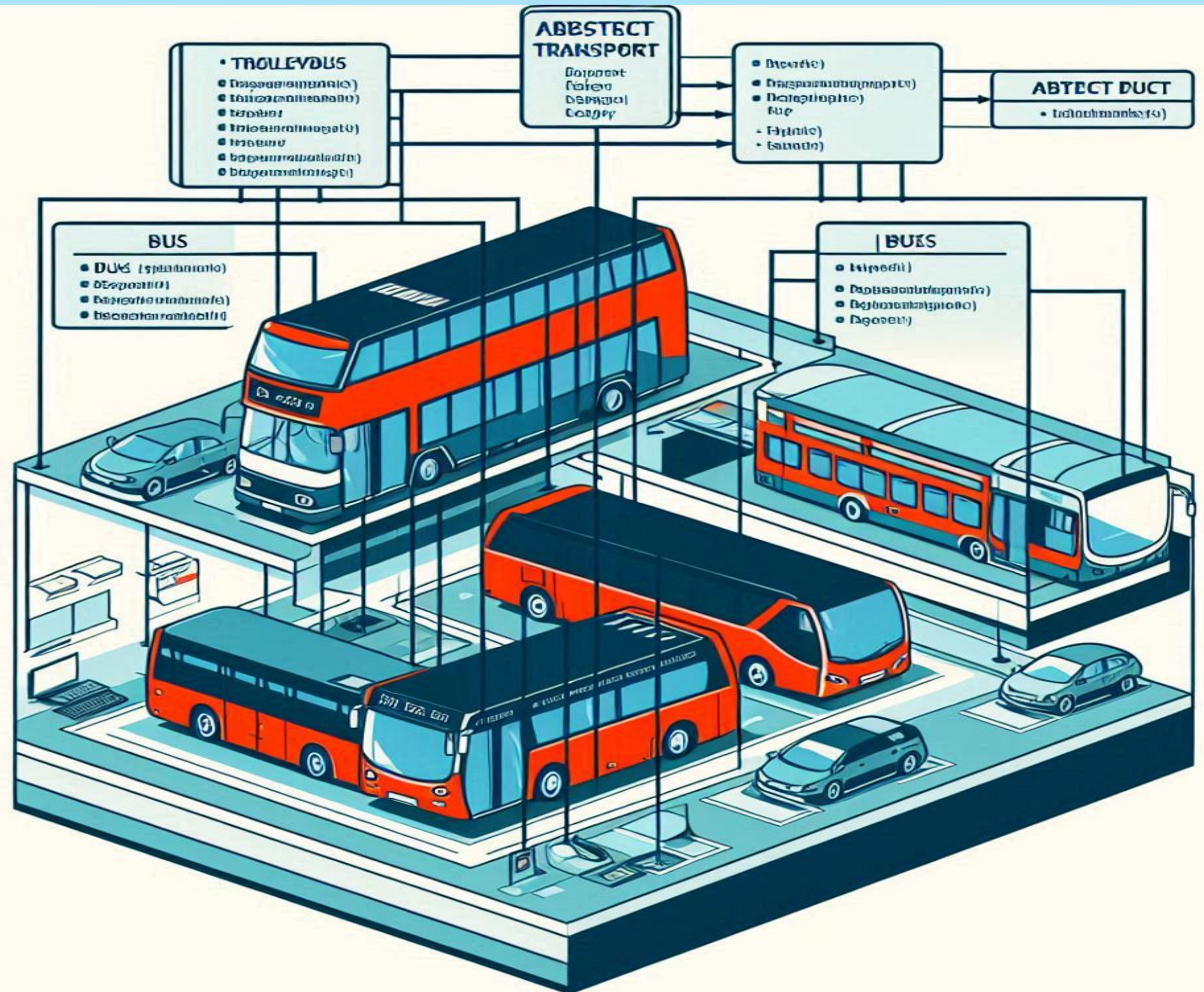
Мұралану

Мысалы, терезеден көшеге қарайтын болсақ “автомобиль”, “автобус”, “үй”, “ағаш”, “троллейбус”, “бұлт”, “адам” сияқты класстардың объектілерін байқаған болар едік. “автобус” класы мен “троллейбус” класы “ағаш” немесе “адам” класымен салыстырғанда бір-біріне жақынырақ екенің байқаймыз. Яғни қасиеттер жиынтығы бір-біріне жақынырақ. Егер басқа адам бізбен салыстырғанда нақтылықтан көбірек абстракциялаған болса, онда мысалы, “автобус”, “троллейбус”, “трамвай” кластарының объектілерін “қоғамдық көлік” класына жатқызатын болады.



Мұралану

Сонда “троллейбус” класына тиісті объектілер “қоғамдық көлік” класына да жататын болып тұр. Егер класстарды объектілер жиыны деп қарастыратын болсақ “троллейбус” жиыны “қоғамдық көлік” жиынына кіреді деп айтар едік немесе “троллейбус” жиыны “қоғамдық көлік” жиынының ішкі жиыны немесе төменгі жиыны болып саналады. Былайша айтқанда “троллейбус” класы “қоғамдық көлік” класының ішкі класы немесе төменгі класы болып саналады.



Абстрактілі класс

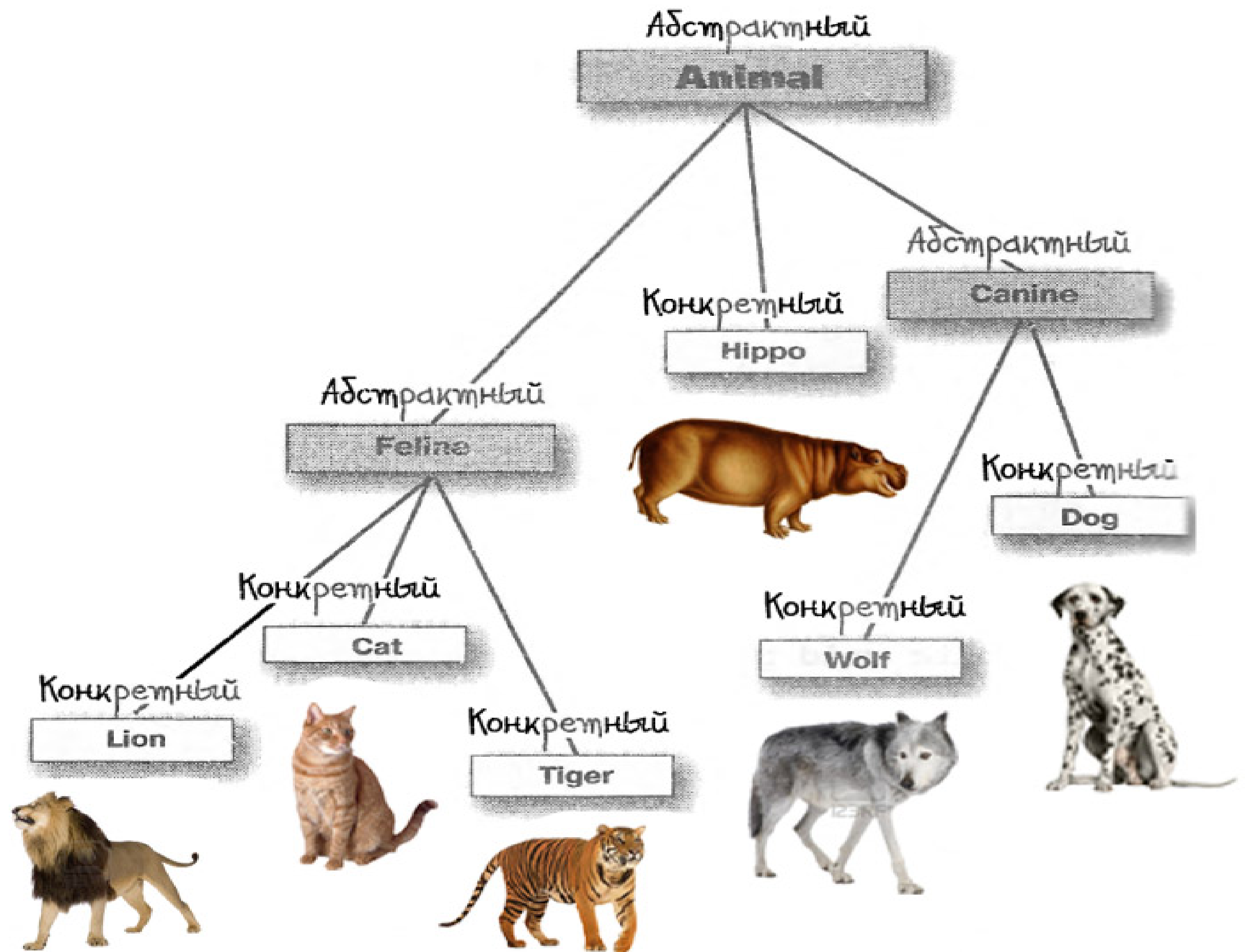
Абстрактілі класс деп объектілері “ұрпақ” класстарының біріне тиісті классты айтамыз.

А класы В класының төменгі класы болып саналады, егер А класына тиісті кез-келген объект В класына да тиісті болса. Онда А класы туынды класс немесе еншілес класс, ал В класын аталық класс деп атаймыз. А класын ұрпағы, ал В класын атасы деп атаймыз. В класын көбірек абстракцияланған класс, ал А класын нақтырақ класс деп атасақ болады.

Әрбір нақты объект өзінің “туған” класына тиісті, бірақ тек “туған” класстың ғана данасы емес, “туған” класстың барлық аталарының да данасы болып саналады. “Туған” классты нақты класс деп, ал барлық аталарын абстрактілі класстар деп атаймыз. Иерархия бойынша көтерілген сайын көбірек абстрактілі болып келеді. Абстрактілі класс деп объектілері “ұрпақ” класстарының біріне тиісті классты айтамыз. Мысалы, “құс” класын абстрактілі класс деп атаймыз, себебі қандай да бір объект “құс” класының да данасы, мысалы, “кептер” немесе “қарға” класының да данасы болатын болса.

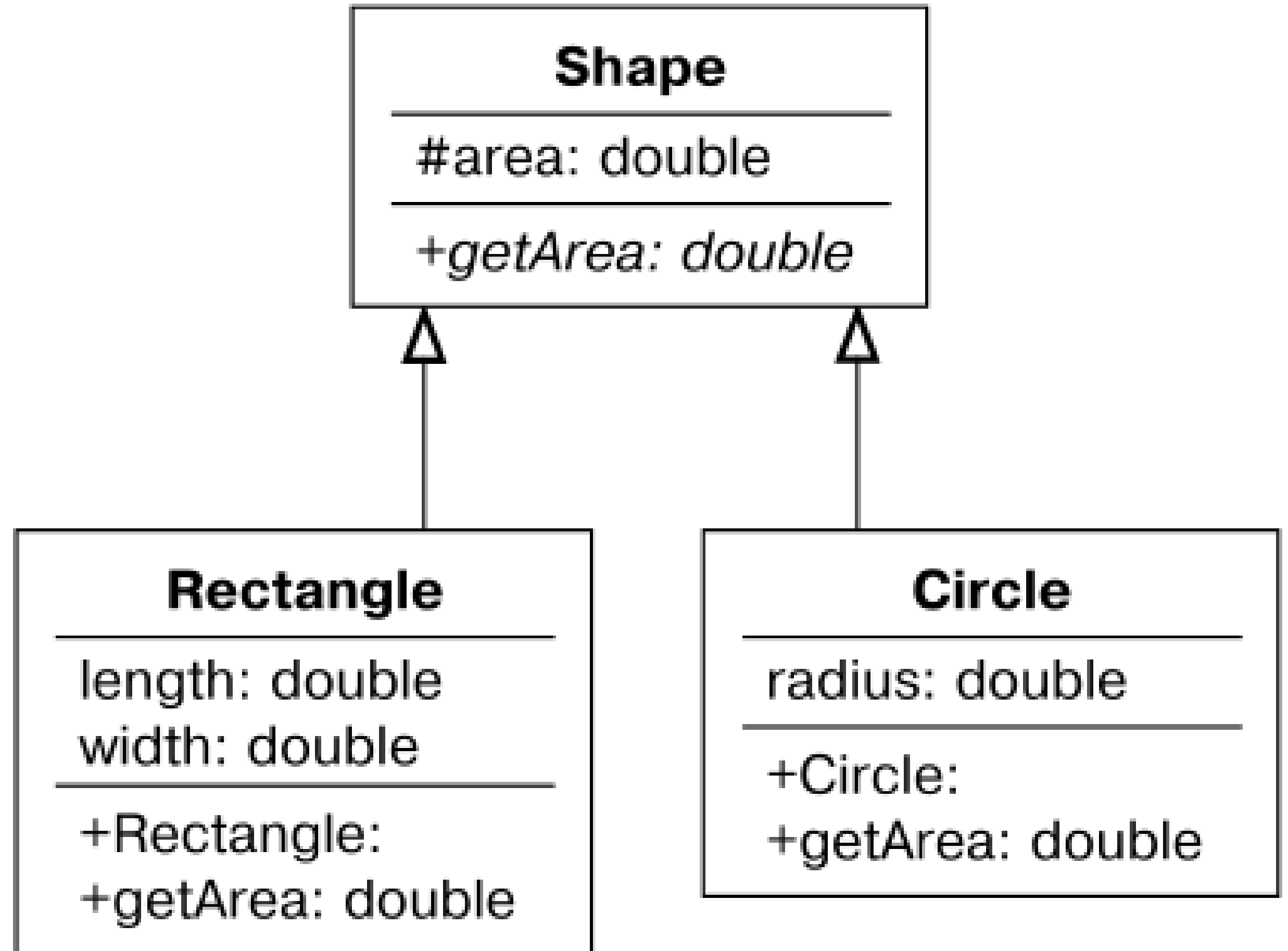
Абстрактілі класс

Абстрактілі класс деп объектілері “ұрпақ” класстарының біріне тиісті классты айтамыз.



Абстрактілі класс

Абстрактілі класс деп объектілері
“ұрпақ” класстарының біріне
тиісті классты айтамыз.



Абстракттілі класс

Абстракттілі класс деп объектілері
“ұрпақ” класстарының біріне
тиісті классты айтамыз.

```
abstract class Shape{ 2 usages 2 inheritors  
  
    protected double area ; no usages  
  
    public abstract double getArea();  
}
```

Абстрактілі класс

Абстрактілі класс деп объектілері
“ұрпақ” класстарының біріне
тиісті классты айтамыз.

```
class Rectangle extends Shape{ no usages
    private double width ; 2 usages
    private double height ; 2 usages

    public Rectangle(double width, double height){
        this.width = width ;
        this.height = height ;
    }

    public double getArea(){ no usages
        return width * height ;
    }
}
```

Абстракттілі класс

Абстракттілі класс деп объектілері
“ұрпақ” класстарының біріне
тиісті классты айтамыз.

```
class Circle extends Shape{ no usages
    private double radius ; 3 usages
    public Circle(double radius){ no usages
        this.radius = radius ;
    }
    public double getArea(){ no usages
        return Math.PI * radius * radius ;
    }
}
```

Абстракция-1

Абстракция ұғымының бірінші мағынасы қолданып жатқан контекстте маңызды деген қасиеттер мен іс-әрекеттерді таңдай отырып классты сипаттау

ОБП құру барысында біздер программада шың өмірдің объектілерін қолдануға тура келеді. Бірақ программада шың өмірдегі объектілердің өзін қолдана алмаймыз, біздер оның аналогын, моделін қолданамыз. Моделін құру барысында біздер объектінің барлық қасиеттір мен іс-әрекеттерін ала алмаймыз. Біз тек қолданып жатқан контекстте маңызды деген қасиеттері мен іс-әрекеттерін қажет етеміз.

Абстракция - бұл шың өмірден алынған бір объектінің немесе құбылыстың кейбір маңызды емес бөліктерін алмай немесе дәл осы контексте маңызы жоқ бөліктерін қалдыра құрылған модель.

Нақты объектінің маңызды деген қасиеттер жиынтығын қалдырып, маңызды емес деген қасиеттерді елемей қарапайым баламасын жасау.

Абстракция-2

Қажет емес мәліметтерді жасыру арқылы күрделілікті басқару.

Бұл іске асыру мәліметтерін жасыру және тек функционалдылықты көрсету процесі.



Инкапсуляция

көп мағыналы сөз. Бірнеше
мағынасын ашып көреміз.

Бір бүтін нәрсені сипаттауға қатысты барлық ақпарат бірге сақталуы керек. Яғни бір болмысқа қатысты ақпараттарды біріктіру.

Бір болмысқа, яғни классқа, деректер мен оны өңдейтін функцияларды біріктіру. Деректер және оны өңдейтін код бір-бірімен тығыз байланыста, сол себепті бір бүтін нәрсеге бірігуі тиіс. Деректермен жұмыс істейтін барлық функциялар бір класстың ішінде метод болып анықталуы керек.

Объектінің ішкі күйлері мен іс-әрекеттерінің бір бөліктерін сыртқы әлеммен, яғни басқа объектілермен, тек белгілі бір интерфейс арқылы байланыс жасауды ұсыну арқылы жасыра алу қабілеті.

Объектінің ішкі жүзеге асыруларын, ішкі механизмдерін, яғни ішкі деректерді күтпеген өзгерістерден сақтау.

Инкапсуляция vs Абстракция

инкапсуляция ішкі деректерді қорғау және бүтіндігін қамтамасыз ететін болса, абстракция объектімен әрекеттесуді, байланысуды, қолдануды жеңілдетеді.

Инкапсуляция – бұл public, private сияқты модификаторларды қолдана отырып ішкі деректерді сыртқы ортадан қорғау. Инкапсуляция тек деректерді жасырып қана қоймайды, ол деректерге қолжетімділікті басқара отырып олардың бүтіндігіне жауап береді.

Абстракция – бұл абстрактілі классты немесе интерфейсті қолданып және ішкі жүзеге асыруларды жасыра отырып объектімен байланысты жеңілдету. Абстракция нақтылықтан қашықтап, жалпы ұғымдарға шоғырлануға мүмкіндік береді. Абстракция төменгі деңгейлі нақтылықтан бас тартып негізгі функционалдылыққа назар аудару арқылы күрделі жүйелерді жобалауды жеңілдетеді.

Инкапсуляция класс құрылымындағы деректер мен ішкі кодтарды жасыру және деректердің бүтіндігін қамтамасыз етумен айналысатын болса, абстракция класстың басқа “клиент” кодта қалай көрінетіндігіне жауап береді, яғни сыртқы көрінісіне, қолдануы ыңғайлылығына.

Инкапсуляция vs Абстракция

инкапсуляция ішкі деректерді қорғау және бүтіндігін қамтамасыз ететін болса, абстракция объектімен әрекеттесуді, байланысуды, қолдануды жеңілдетеді.

```
class EncapsulationDemo {  
    private int number;  
    public int getNumberValue() {  
        return number;  
    }  
    public void setNumberValue(int value) {  
        if(value >= 100)  
            this.number = value;  
    }  
}
```

```
public abstract AbstractionDemo {  
    public abstract void performAction();  
}
```

```
public class ImplementAbstractionDemo extends AbstractionDemo {  
    public void performAction() {  
        // Мұнда бір іс-әрекеттерді орындайды  
    }  
}
```

Абстракция

Қажет емес мәліметтерді жасыру арқылы күрделілікті басқару.

Абстракция - бұл шың өмірден алынған бір объектінің немесе құбылыстың кейбір маңызды емес бөліктерін алмай немесе дәл осы контексте маңызы жоқ бөліктерін қалдыра құрылған модель.

Нақты объектінің маңызды деген қасиеттер жиынтығын қалдырып, маңызды емес деген қасиеттерді елемей қарапайым баламасын жасау.

Қажет емес мәліметтерді жасыру арқылы күрделілікті басқару. Классты жобалаушыға клиент жағынан қарағанда ыңғайлы интерфейсті құруға көмектеседі.

Бұл іске асыру мәліметтерін жасыру және тек функционалдылықты көрсету процесі.

Бұл объектілердің немесе жүйенің жалпы сипаттамалары мен функционалдығын бөліп көрсете отырып, іске асыру мәліметтерін елемей процесі.

Объектілердің жалпы сыртқы сипаттамаларын ғана пайдалана отырып, олардың нақты егжей-тегжейлі жүзеге асырылу сипаттамаларын елемей қолдану.

Қорытынды

01



Абстракция ұғымының
мағыналарын кеңірек
ашып қарастырдық

02

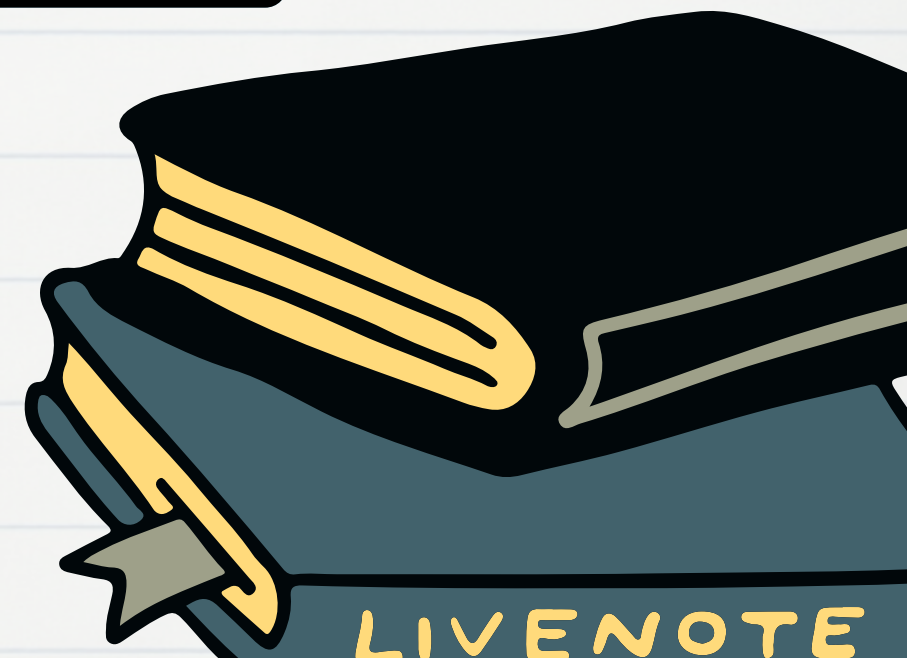


Мұралау принципімен
таңыстық

03



Абстракция және
инкапсуляция
ұғымдарының
айырмашылығын
көрсеттік



Назарларыңызға рақмет

