

ЛЕКЦИЯ I © 2020 Нет Ит

# JAVA OOP: Наследяване

Теодор Костадинов



SOFTWARE  
ACADEMY



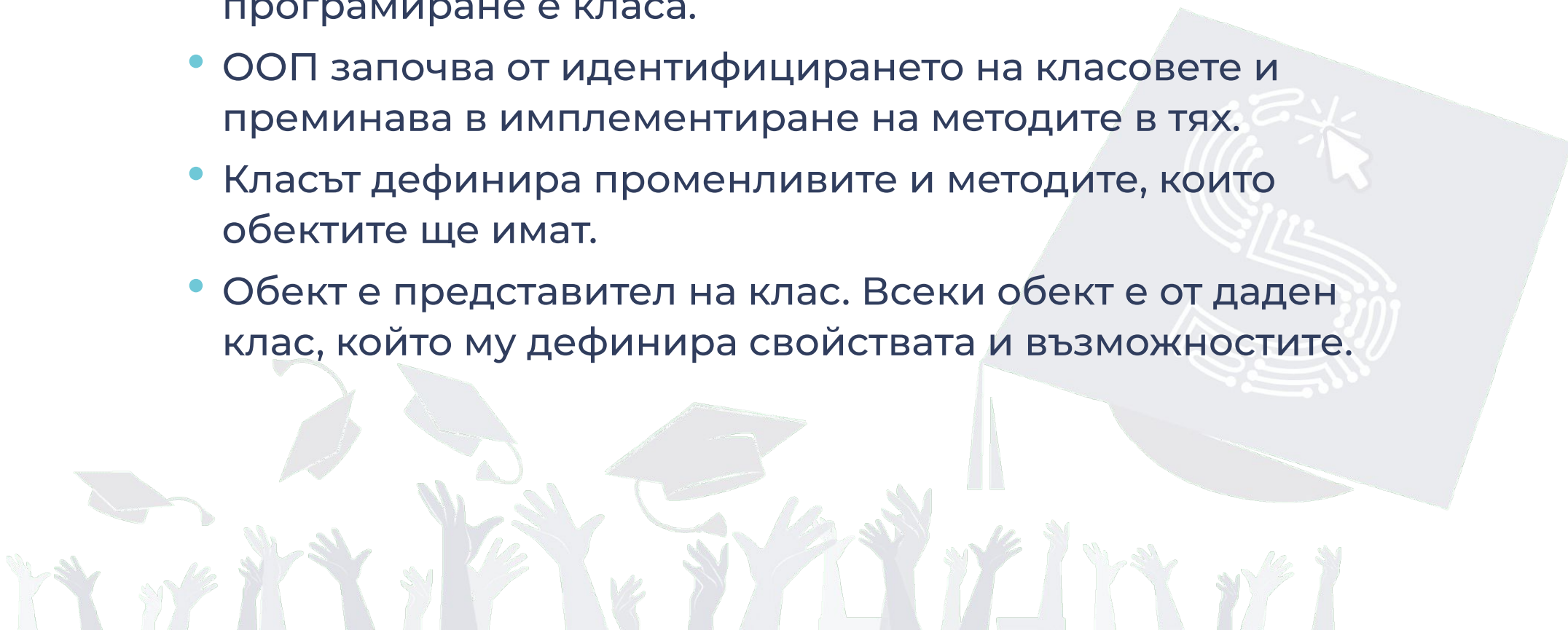
# Съдържание

1. Статичност
2. Енумерация
3. Наследяване
- 4.



# Какво е ООП?

- Основна единица на обектно ориентираното програмиране е класа.
- ООП започва от идентифицирането на класовете и преминава в имплементиране на методите в тях.
- Класът дефинира променливите и методите, които обектите ще имат.
- Обект е представител на клас. Всеки обект е от даден клас, който му дефинира свойствата и възможностите.



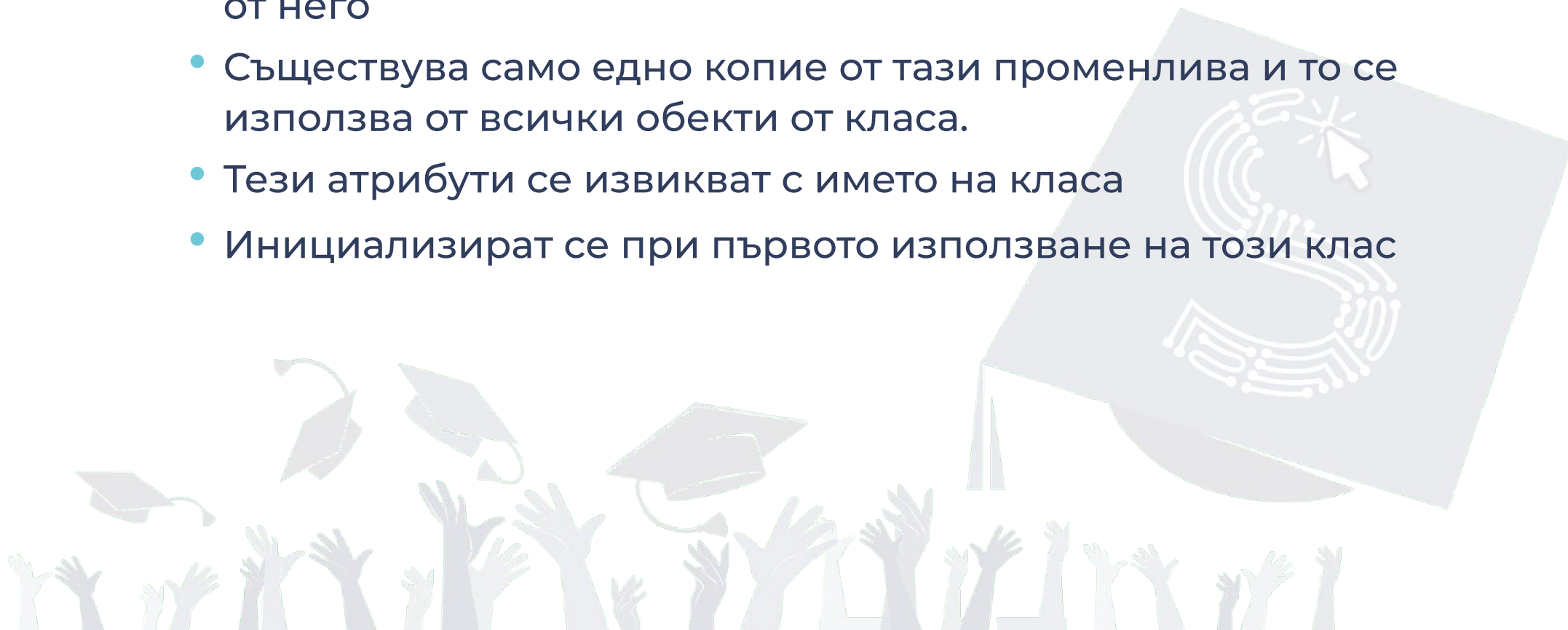


# Статичност



# Какво е статичен член?

- Дадено поле или метод се асоциират с класа, а не с обект от него
- Съществува само едно копие от тази променлива и то се използва от всички обекти от класа.
- Тези атрибути се извикват с името на класа
- Инициализират се при първото използване на този клас



- Нека при създаването на обект куче от класа Куче - на всяко новосъздадено куче да се задава уникален пореден номер

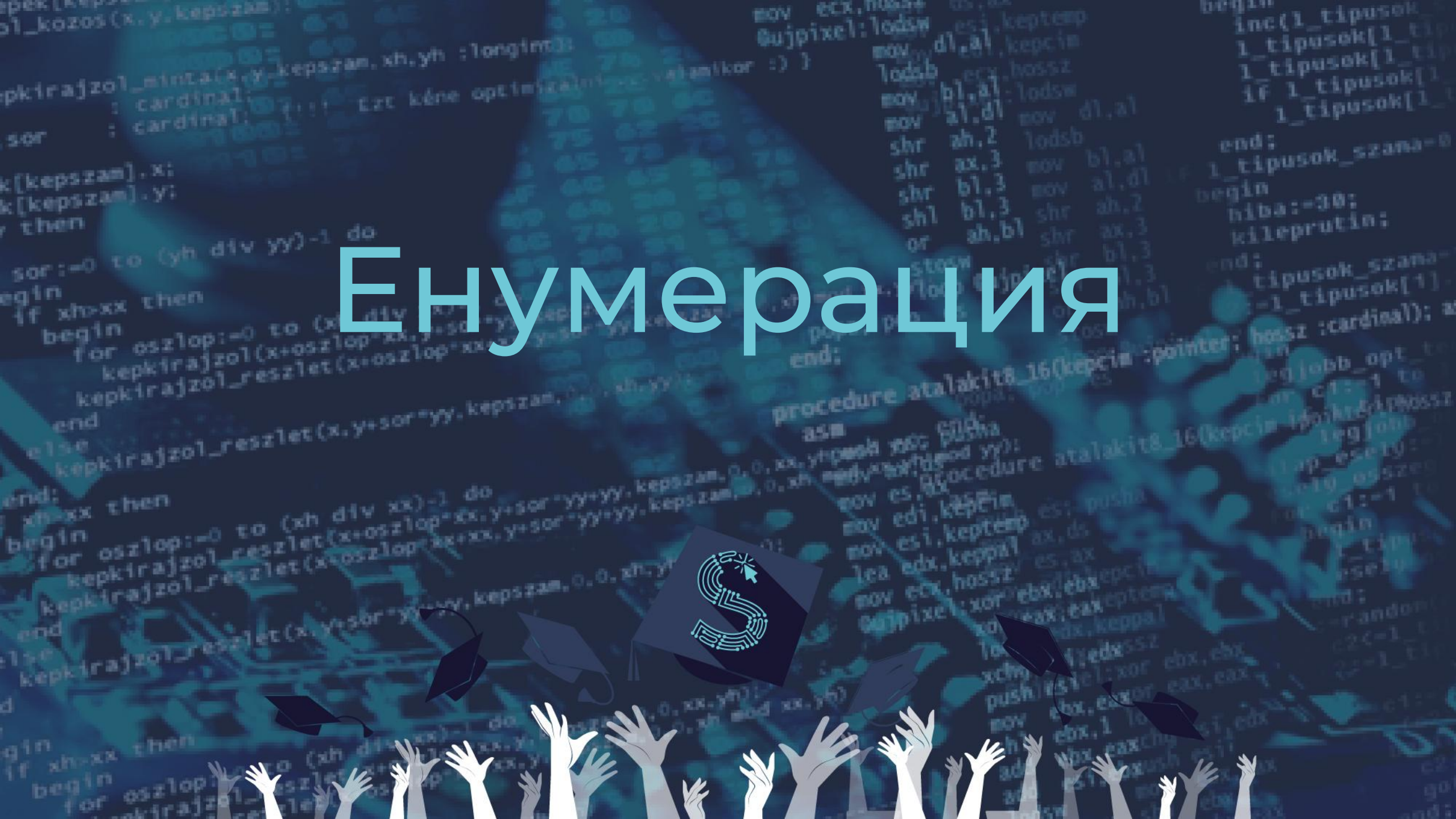


Ами сега?

Задача за упражнение



# Енумерация



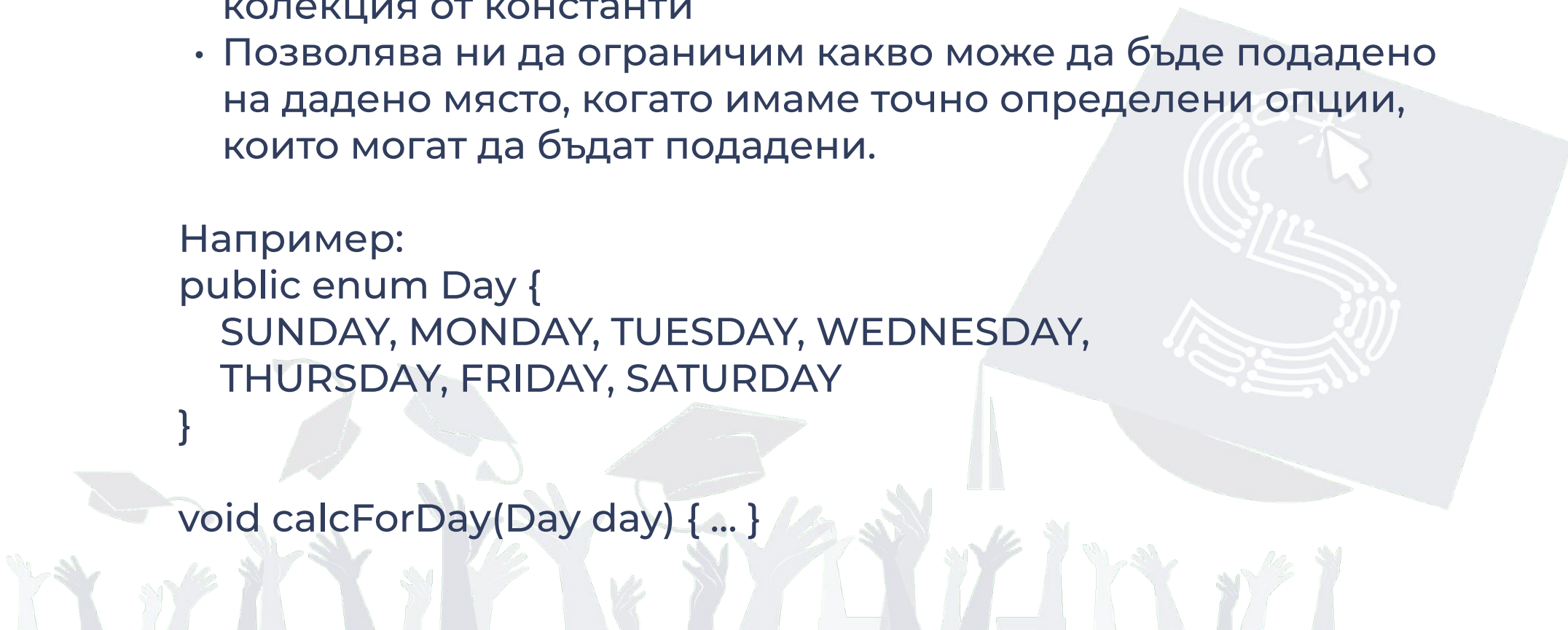
# Какво е enum?

- Специален тип данни, които се използват, когато имаме колекция от константи
- Позволява ни да ограничим какво може да бъде подадено на дадено място, когато имаме точно определени опции, които могат да бъдат подадени.

Например:

```
public enum Day {  
    SUNDAY, MONDAY, TUESDAY, WEDNESDAY,  
    THURSDAY, FRIDAY, SATURDAY  
}
```

```
void calcForDay(Day day) { ... }
```



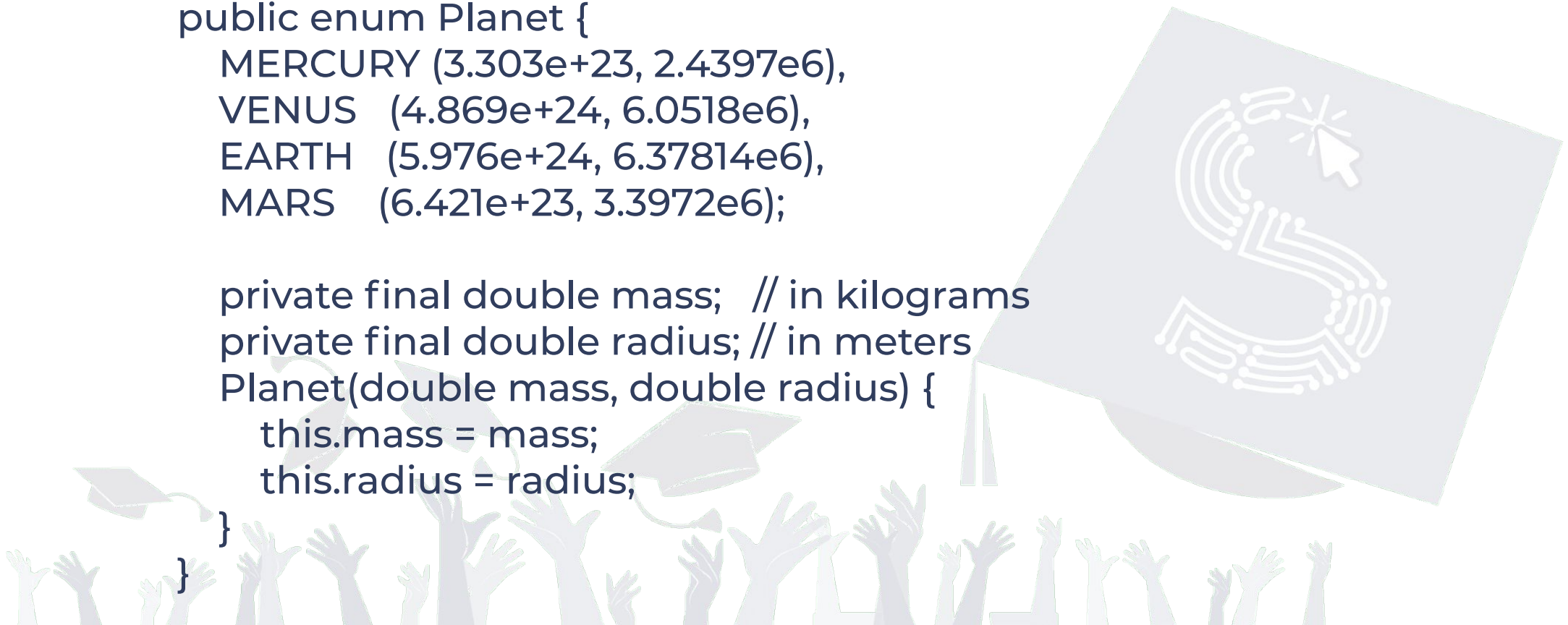


# Enum блягинки

- Можем да вземем всички стойности в един enum.
  - `Day.values()`
- Можем да добавяме допълнителни стойности в enum-а

```
public enum Planet {  
    MERCURY (3.303e+23, 2.4397e6),  
    VENUS   (4.869e+24, 6.0518e6),  
    EARTH   (5.976e+24, 6.37814e6),  
    MARS    (6.421e+23, 3.3972e6);  
}
```

```
    private final double mass; // in kilograms  
    private final double radius; // in meters  
    Planet(double mass, double radius) {  
        this.mass = mass;  
        this.radius = radius;  
    }  
}
```



# Принципи на ООП

---

- **Наследяване**
- **Полиморфизъм**
- **Абстракция**
- **Капсулация**

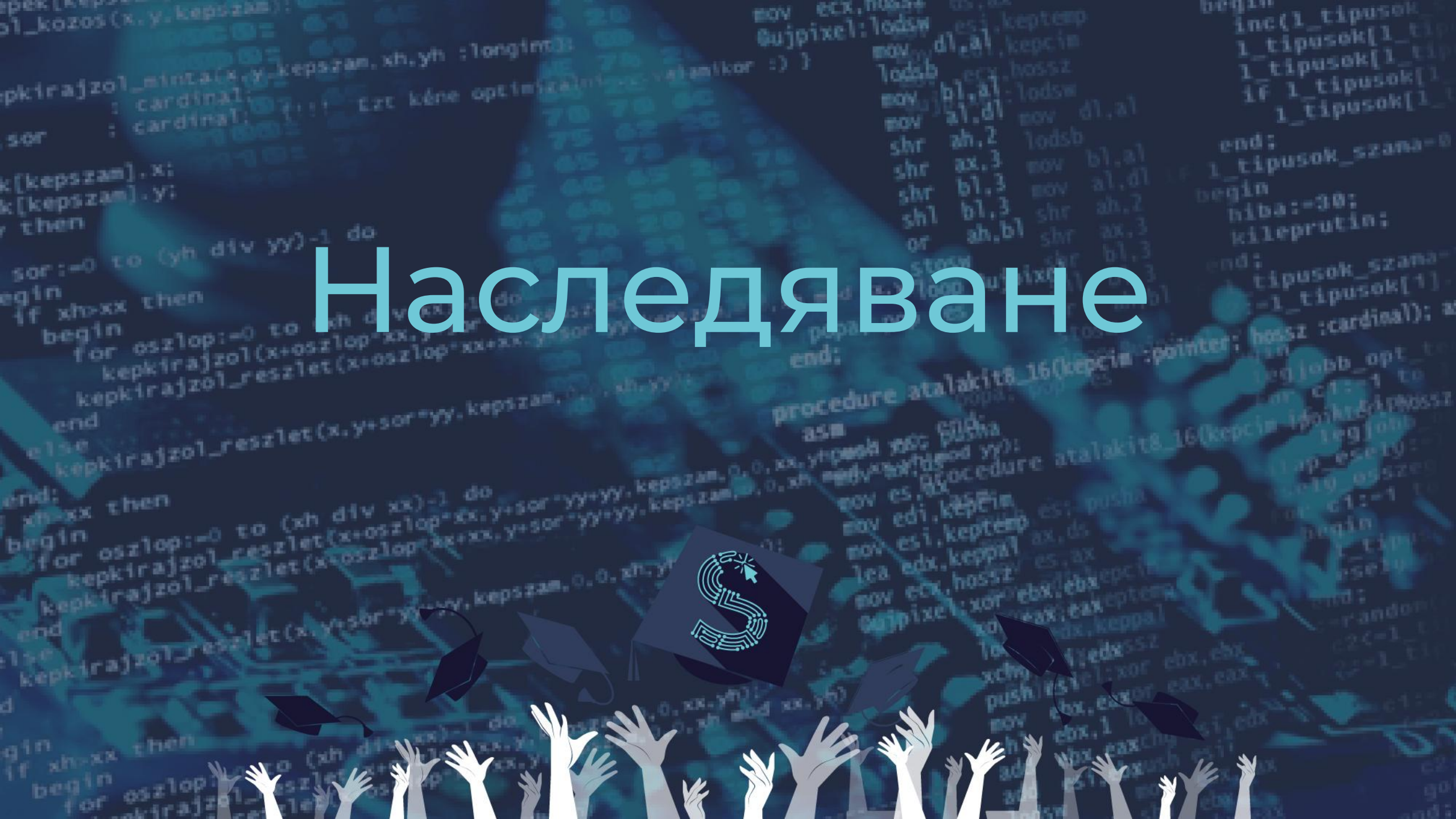


до 19:35



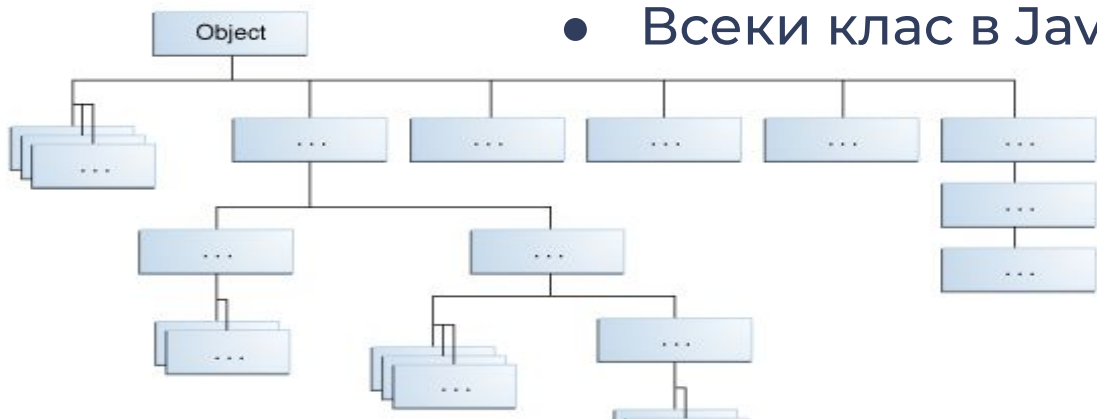


# Наследяване



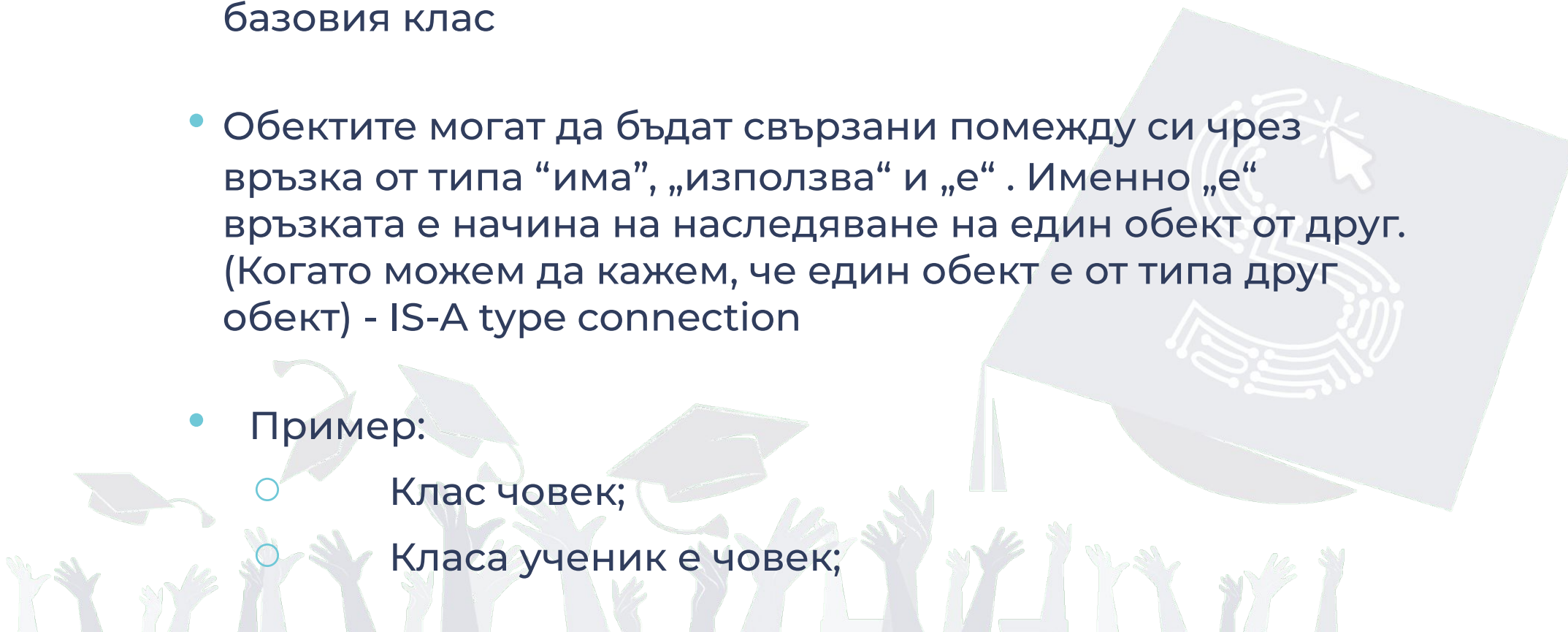
# Наследяване

- Наследяването е процес по създаване на клас, който е под-клас на друг.
- Новосъздаденият клас се нарича под-клас, клас-дете, extended class, etc
- Класът, от който се наследява, се нарича базов клас, родителски клас, наследен клас
- Всеки клас може да има само един родителски клас.
- Класът-дете наследява всички полета и методи на родителя си.
- Всеки клас в Java по дефолт наследява класа Object

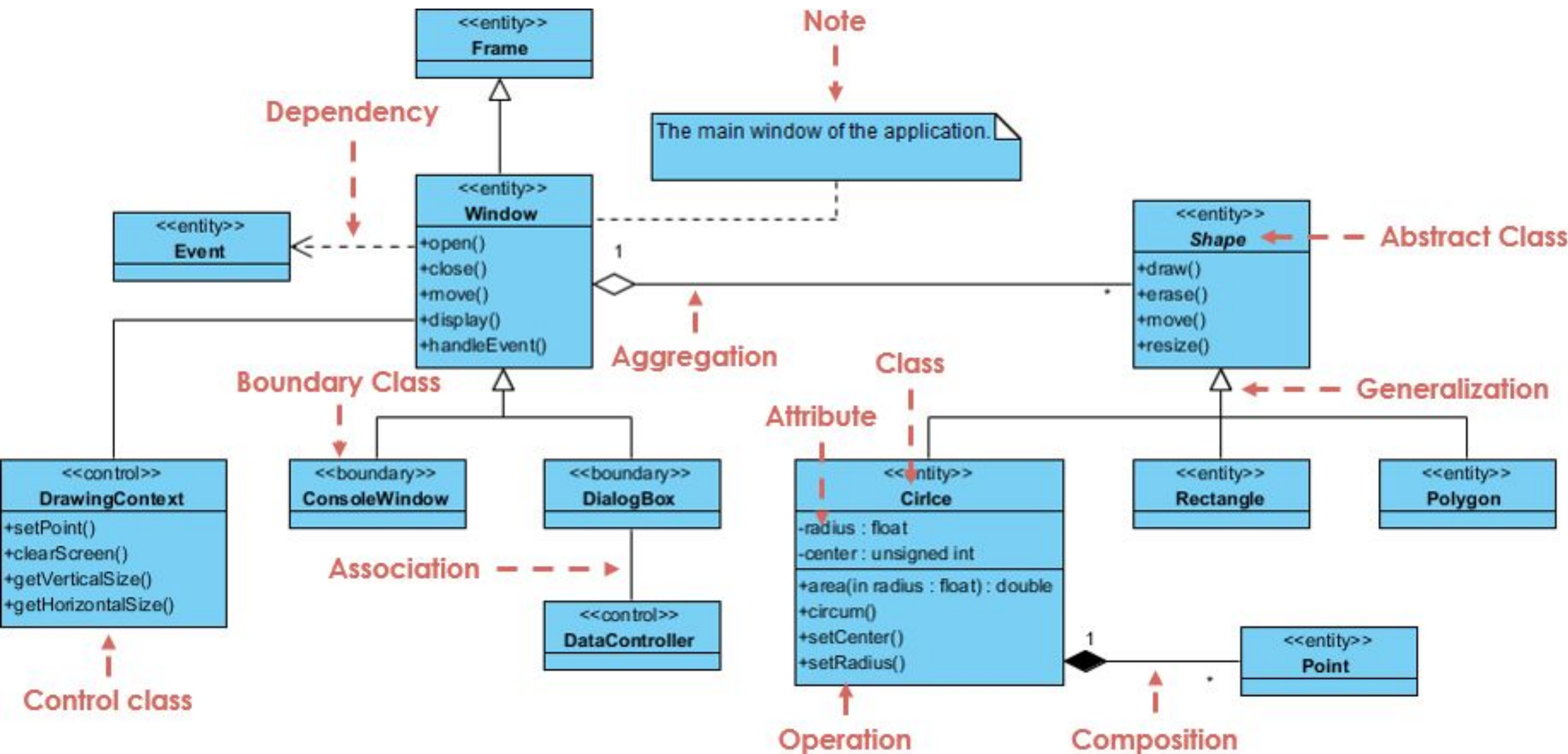


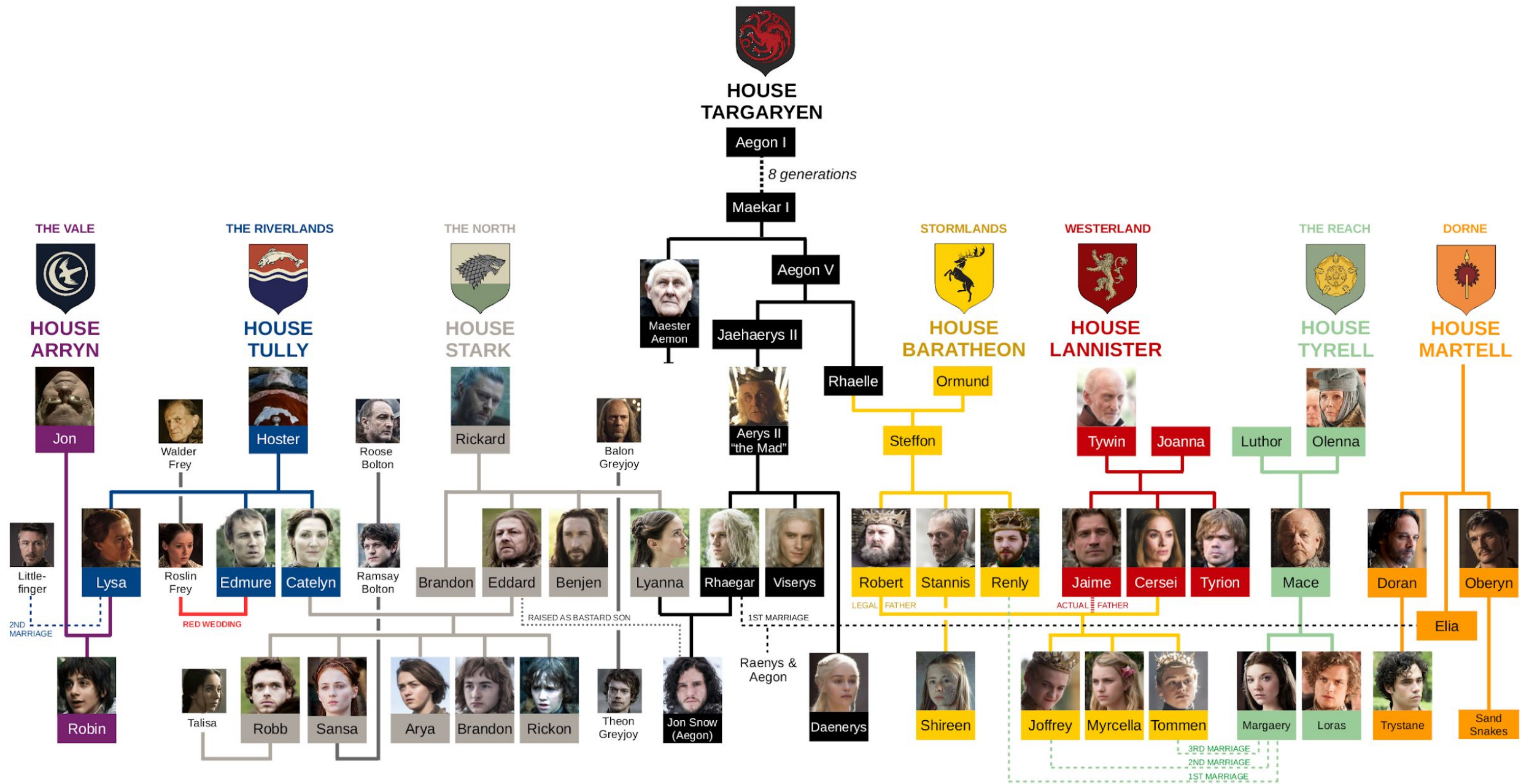
# Наследяване

- Една от най-силните черти на наследството е възможността за extend-ване на компоненти без да се знае нищо за начина, по който са имплементирани в базовия клас
- Обектите могат да бъдат свързани помежду си чрез връзка от типа “има”, „използва“ и „е“. Именно „е“ връзката е начина на наследяване на един обект от друг. (Когато можем да кажем, че един обект е от типа друг обект) - IS-A type connection
- Пример:
  - Клас човек;
  - Класа ученик е човек;









# В един клас може да намерите

- Полета: променливи, които определят настоящия статус на обекта
- Статични полета: променливи, които са общи за всички обекти от класа и по-скоро определят статуса на класа като цяло, не на отделните обекти от този клас.
- Методи: изпълним код, позволяващ ни да променяме състоянието на обекта или да достъпваме данни от него
- Статични методи: изпълним код, отнасящ се за класа като цяло, не трябва да използва в себе си променливи, които не са статични
- Вложени класове и интерфейси





# Оператор за проверка на типа

- instanceof
  - Използва се за да се провери дали даден обект е от определен тип/клас

```
if(cat instanceof Dog) {  
    ((Dog) cat).bark();  
}
```



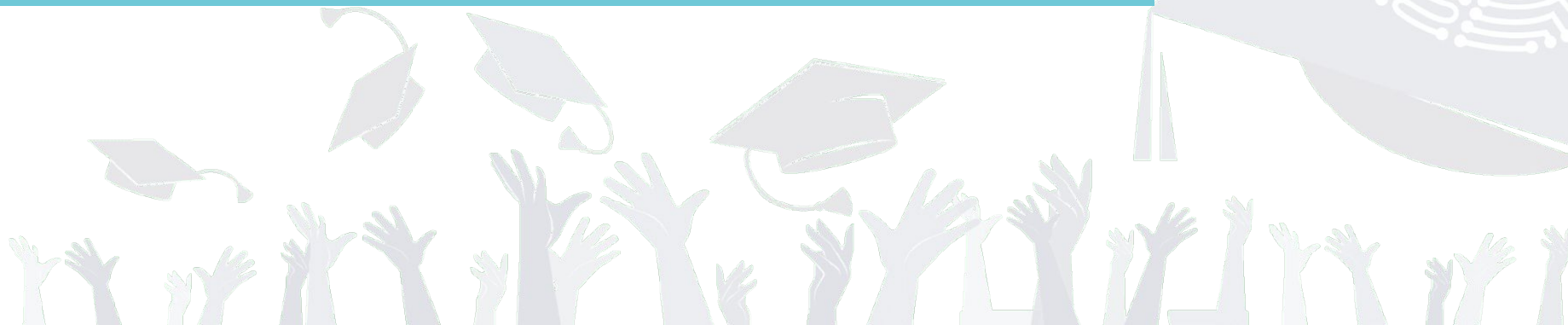
ВЪПРОСИ?



# Резюме



- Наследяването помага да се избегне повторение на код
- Може да се наследява само един клас, трябва да се избегне нуждата от множествоно наследяване.





# Ресурси

- [Docs](#)
- [GitHub Repo with Demos](#)
- [Class Diagram Article](#)
- [Head First Java](#)



# Задачи за упражняване



# Задача 1

Създайте йерархия Dog, Frog, Cat, Kitten, Tomcat и дефинирайте съответните конструктори и методи за всеки клас. Кучетата, жабите и котките са Animals. Всички животни могат да издават звуци. Kittens и tomcats са котки. Всички животни имат години, име и пол. Kittens могат да бъдат само женски, а tomcats – мъжки. Всяко животно издава специфичен звук.

\*Създайте масиви от различни видове животни и пресметнете средната възраст на всеки вид животно използвайки статичен метод.



Ами сега?

Задача за упражнение



# Задача 2

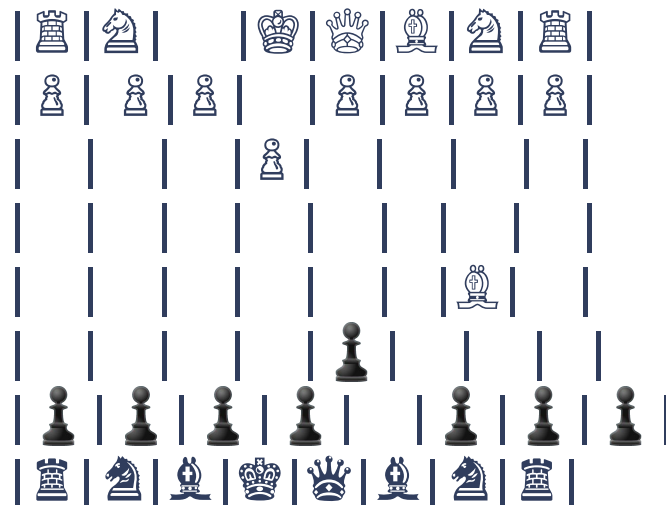
Създайте класове за всички шахматни фигури и имплементирайте проверката за правилен ход на всички  
Добавете в класа `PlayingPiece` поле цвят  
Създайте клас дъска, който при създаването си да инициализира двумерен масив 8x8 и в него да слага фигурите



Ами сега?

Задача за упражнение

# Задача 2



Тук са кодовете на символите  
[http://www.utf8-chartable.de/unicode-utf8-table.pl?  
start=9728&number=128](http://www.utf8-chartable.de/unicode-utf8-table.pl?start=9728&number=128)



Ами сега?

Задача за упражнение



Това домашно влиза в крайната ви оценка!

# Домашно

Качвайте домашното си в ГитХъб и  
слагайте линка тук:

<https://forms.gle/AcvCptCbSDizr2Ay6>





# Задача 1



Видеотека:

Създайте система за вземане на видеокасети от видеотека.

1. В системата има два типа акаунт - админ и клиент. И двата акаунта имат име, имейл и парола. И двата акаунта имат опция логат.
2. За да се влезе в системата, трябва да се въведе имейл и парола.
3. Клиентите могат да проверят кои касети са взели. Могат да върнат взета касета.
4. Клиентите могат да видят всички свободни касети. Могат да вземат касета.
5. Админа може да се създава акаунт на клиент. Админският акаунт е само един и е хардкоднат.
6. Админите могат да видят всички клиенти в системата. Админите могат да видят всички касети.
7. Админите могат да видят всички взети касети и кой ги е взел.

## Задача 2 (незадължително)

Класа дъска да има метод `play(int color, int fromX, int fromY, int toX, int toY)` - да не разрешава да се мести ако два пъти по ред се играе един и същ цвят

Класа дъска да има проверка дали на дъската има шах или шахмат, тя да се извиква след всеки ход

След всеки ход дъската да се чертае на конзолата



© 2020 Нет Ит

# БЛАГОДАРЯ ЗА ВНИМАНИЕТО!



SOFTWARE  
ACADEMY

