

# МЕТОДИ НА КЛАСОВЕ

ЛЕКЦИОНЕН КУРС “ПРОГРАМИРАНЕ НА JAVA”



# КЪМ МОТИВАЦИЯТА: КЛАС 'TIME' С РАЗШИРЕНА ПОСТАНОВКА НА ЗАДАЧАТА

```
class Time {  
    ...  
    printTime() {  
        //print time (US convention)  
    }  
}
```

Искаме да разширим класа:

- Извод на времената според американска (1:40 PM) или европейска (13:40) конвенция
- Приложната програма може да избира: единен формат за всички времеви обекти (сравни Schedule: 4 time plans)
- Приложната програма може да превключва формата (us -> eu, eu -> us)

# ПЪРВО ПРЕДЛОЖЕНИЕ ЗА РЕШЕНИЕ

1

Изпълнена ли е напълно задачата?

```
class Time {  
    ...  
    private boolean usTime = true;  
  
    public void printTime () {  
        if (usTime)  
            printUSTime();  
        else  
            printEUTime();  
    }  
  
    private void printUSTime () ...  
  
    private void printEUTime () ...  
  
    public void switchTimeFormat () {  
        usTime = ! usTime;  
    }  
}
```

# ПРОБЛЕМИ

```
public static void main (...) {  
    TimeC t1 = new TimeC(8,30);  
    TimeC t2 = new TimeC();  
  
    t1.printTime();  
    t2.printTime();  
    t1.switchTimeFormat();  
    t2.switchTimeFormat();  
    t1.printTime();  
    t2.printTime();  
}
```

- Единен формат за всички времеви обекти не е безусловно гарантиран
- Причина: инстанционната променлива 'usTime' съществува многократно (за всеки обект по веднъж)

променливата 'usTime' трябва да съществува само веднъж:  
**променлива на клас**



# ПРОМЕНЛИВИ: ДВА ВИДА

## Променливи на класове

```
class TimeC {  
  
    private int hour, minute;  
  
    private final static int noonHour = 12;  
  
    private final static int noonMinute = 0;  
  
    private static boolean usTime = true;  
    ...  
}
```

Променливи на класове (static):  
съществуват веднъж за клас

→ за обмен на информация между  
повече обекти от същия клас

## Променливи на инстанции

Променливи за инстанции  
(non-static):  
създават се за всеки обект

→ описват състоянието на  
инстанцията

# СЪЩЕСТВУВАНЕ НА ПРОМЕНЛИВИ

```
class TimeC {  
    private int hour, minute;  
  
    private final static int noonHour = 12;  
    private final static int noonMinute = 0;  
    private static boolean usTime = true;  
    ...  
}
```

Променливи  
на класове

Променливи  
на инстанции

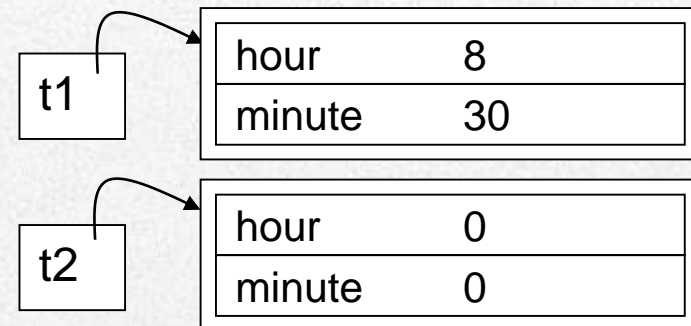
Съществуване в  
началото на програмата

noonHour	12
noonMinute	0
usTime	true

към клас TimeC

Съществуване при  
създаване на един обект:

```
t1 = new TimeC(8, 30);  
t2 = new TimeC(0, 0);
```



към обектите

# СМИСЪЛ НА ПРОМЕНЛИВИТЕ НА КЛАСОВЕ(1):

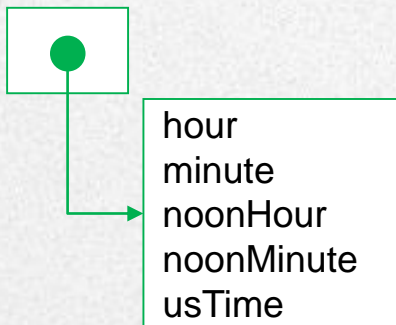
Общата информация за всички обекти се съхранява само веднъж (ефективност на програмната логика)

За сравнение: алтернативно решение

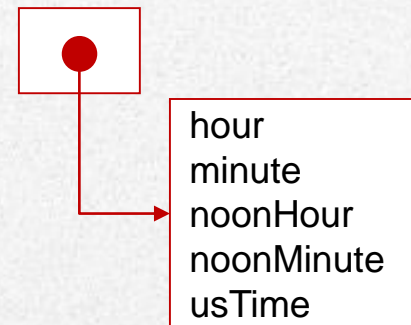
```
class Time {  
    private int hour, minute;  
  
    private final int noonHour = 12;  
    private final int noonMinute = 0;  
    private boolean usTime = true;  
}
```

без static

t1:



t2:



# СМИСЪЛ НА ПРОМЕНЛИВИТЕ НА КЛАСОВЕ(2):

Реализиране на императивно програмиране: не се създават обекти

Данните (променливите) се използват само веднъж

```
class TimePlan {  
  
    static int hour, minute;  
  
    static addMinutes (int m)    {  
        ...  
        hour = totalMinutes / 60;  
    }  
}
```

- променливите hour, minute съществуват от началото на програмата
- няма създаване на инстанции
- стил на програмиране както при Pascal, C – в центъра алгоритми (напр. Quicksort, Hanoi ... там ОО не е смислено)



# МЕТОДИ: ДВА ВИДА

## Методи на инстанции

```
class TimeC {  
    ...    // Variablen
```

```
    public addMinutes (int m)    {  
        int totalMinutes = ...  
        hour = ...;  
        minute =...;  
    }
```

## Методи на класове

```
    public static void switchTimeFormat()    {  
        usTime = !usTime;  
    }  
}
```

Метод на инстанция: свързан с обекта

→ съществува толкова често, колкото пъти се създава инстанцията

Метод на клас: свързан с класа

→ съществува само веднъж

# СМИСЪЛ НА МЕТОДИТЕ НА КЛАСОВЕ

Алгоритъмът (метод) трябва да съществува само веднъж

```
class TimeC {  
    int hour, minute;           // Instance variables  
    static boolean usTime; // Class variables  
  
    public addMinutes (int m)    {  
        int totalMinutes = (60*hour + minute + m);  
    }  
    public static void switchTimeFormat() {  
        usTime = !usTime;  
    }  
}
```

**Методи на клас:** могат да обработват **само** променливи на класове

**Методи на инстанции:** **всички** променливи (на класове и инстанции) могат да бъдат обработвани

# ИЗВИКВАНЕ НА МЕТОДИ НА КЛАСОВЕ:ОТНОШЕНИЕ КЪМ КЛАСА

```
class TimeC {  
    int hour, minute;  
  
    public printTimeInMinuesMinutes()    {}  
    public static void switchTimeFormat()    {}  
}
```

```
class ScheduleC {  
    main (...)    {  
        TimeC t1 new TimeC(8,30);  
        TimeC t2 new TimeC(0,0);  
        int min;  
  
        t2.printTimeInMinutes();  
        TimeC.switchTimeFormat ();  
        ...  
    }  
}
```

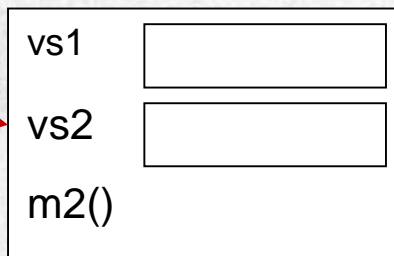
Методи на инстанции: свързани  
към инстанцията

Методи на класове: свързани  
към класа

# ПРОМЕНЛИВИ И МЕТОДИ (ОБОБЩЕНИЕ)

```
class C {  
    int v1, v2;  
    static int vs1, vs2;  
    void m1() {...}  
    static void m2() {...}  
}
```

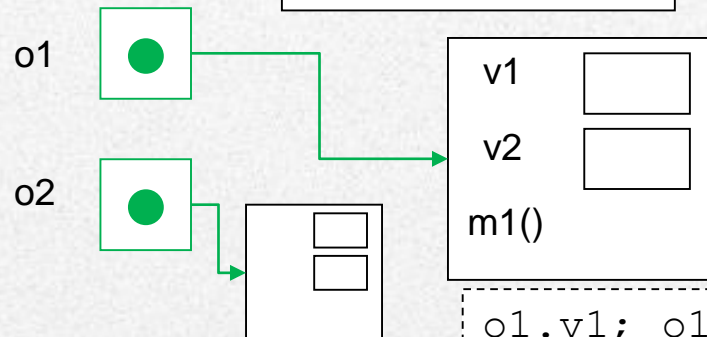
Съществуване в  
началото на програмата  
(само веднъж)



`C.vs1; C.m2();`

Съществуване при  
създаване на обекта (ев.  
многократно)

```
o1 = new C();  
o2 = new C();
```





# MAIN МЕТОД

1 Какъв метод е main?

Винаги на клас

БЛАГОДАРЯ ЗА ВНИМАНИЕТО!

КРАЙ “МЕТОДИ НА КЛАСОВЕ”

