# Лекция 23
# Конструктори. Прототипи.

**Задача 1:**
Да се създадат конструкторите Task и Employee.

Конструкторът Task представя някакво количество работа(задача) което трябва да бъде свършено.
Конструкторът да дефинира и приема следните полета:

- name
- workingHours – показва колко часа остават на задачата за да бъде изпълнена

Да се създаде конструктор за Employee създаващ обект-работник.
Нека да има следните полета:

- name – име на работника
- currentTask – обект който показва текущата задача в/у която работи работника
- hoursLeft – числова стойност, която показва колко работни часа остават на работника (за днес)

Да се направи метод без параметри work() на Employee.
С извикването на този метод, работника за който е извикан метода работи по текущата си задача (ако има такава). Например ако работника има останали 4 часа за работа, а по задачата има останали 7 часа работа, след изпълнението на метода работника ще е останал с 0 часа работа за деня, а по задачата ще е останало 3 часа работа. Ако работника има 8 часа, след изпълнението на метода work() работника ще е останал с 1 час работа за деня, а времето което остава на задачата да бъде свършена ще е 0.

Да се направи и метод showReport, който се извиква след като работника поработи в/у текущата си задача (извикване на work) и показва информация (принтирайки на конзолата) за:

- името на работника
- името на задачата
- работните часовете които остават на работника

- часовете които остават на текущата задача на работника за да бъде изпълнена

Да се тества цялата функционалност.

## Задача 2:

Допълнете задачата от час за Computer да има следния конструктор с параметри :

Computer(year, price, isNotebook, hardDiskMemory, freeMemory, operationSystem)

който инициализира всички полета със стойностите подадени като аргументи на конструктора.

-метод comparePrice(computer), който сравнява цените на 2 компютъра. Ако цената на първия компютър (този, за който се вика метода) е по-висока от тази на компютъра, подаден като аргумент, методът връща -1. Ако цената на компютъра подаден като аргумент е по-високата, се връща 1, а ако са равни, методът връща стойност 0.

Да се създадат няколко обекта Computer.

Да се сравнят цените на някои от компютрите (посредством метода comparePrice) и да се изведе подходящо съобщение.

## Задача 3:

Създайте конструктор Student, създаващ студенти.

Нека да приема следните данни:
name – име на студента
subject - специалност
grade – успех
yearInCollege – курс
age – години на студента
hasDegree – дали е завършил или не
money – пари от стипендии.

Нека да се дефинират следните методи:
-метод upYear(), който увеличава годината в колежа с единица, ако студента не е завършил, а в противен случай извежда подходящо съобщение. Ако след увеличаването, годината стане 4, задава true на полето hasDegree

-метод receiveScholarship(min, amount) – добавя сумата amount към парите на студента, само ако успехът му е по-висок или равен на минималния успех, подаден като параметър (min) и ако възрастта му е под 30 години. Методът връща текущите пари на студента (след евентуалното им увеличение).

Да се създаде и конструктор StudentGroup(groupSubject), създаващ обект представящ група от студенти от една и съща специалност.

Полета на новосъздадения обект:

groupSubject – специалност на студентите в групата (от параметъра)
students – студенти в групата(масив от обекти)
freePlaces – свободни места в групата.

Конструктора да създава места за 5 студента в групата (инициализира полето students с нов масив от 5 елемента) и задава стойност 5 на полето freePlaces.

Да се дефинират следните методи:

-метод addStudent(student), който добавя студент към групата ако специалността на студента съвпада с тази на групата и разбира се, ако има свободни места.
След добавяне на студент към групата, да се намалят свободните места в групата.

-метод emptyGroup(), който освобождава всички места в групата и задава стойност 5 за полето freePlaces.

-метод theBestStudentName() - Връща името на студента с най-висок успех в групата.

-метод printStudentsInGroup() - Изкарва информация (Име, успех, ...)за всички студент в курса.
Да се демонстрира използването на класовете Student и StudentGroup (Създават се няколко студента, добавя  се стипендия на някои от тях, прехвърлят се някои от тях в по- горна година. Създадете няколко групи от студенти, добавете студенти в тях, изкарайте името на най- добрия студент от някоя от групите.)