

Наследяване. Final и Abstract.

Задача 1:

Task/Employee част 2 – продължение на домашното от лекция 14

Целта е да се състави симулация на работа в офис. В офиса работят работници (Employee) по определени задачи (Task)

Да се използват класовете Employee и Task част 1

Класът Task остава непроменен:

```
Task
  name : String
  workingHours : int
  Task(String, int)
  getName() : String
  getWorkingHours() : int
  setWorkingHours(int) : void
```

Да се създаде клас AllWork, които репрезентира множеството от всички задачи, които трябва да свършат работниците в офиса.

```
AllWork
  tasks : Task[]
  freePlacesForTasks : int
  currentUnassignedTask : int
  AllWork()
  addTask(Task) : void
  getNextTask() : Task
  isAllWorkDone() : boolean
```

Класът да съдържа следните полета:

- tasks – масив, съдържащ всички задачи
- freePlacesForTasks – числова стойност, която показва колко свободни места за нови задачи има в масива (след създаване на обекта AllWork, в него няма добавени задачи и тази стойност е 10)

–currentUnassignedTask – указва индекса на текущата свободна(по която няма работник които да работи) задача.

Всяка задача може да бъде давана на точно 1 работник, които да работи по нея. Когато някои работник си свърши задачата, той си взима от обекта AllWork първата свободна задача. Това поле указва индекса на текущата такава задача в масива tasks (началната му стойност е 0 – първата задача)

–Да се направи конструктор без параметри, който инициализира по подходящ начин полетата на AllWork

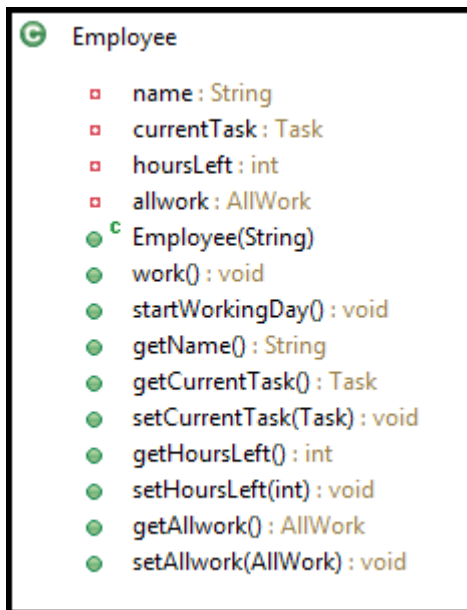
–метод addTask, чрез който се добавя задача към множеството от задачи

–метод getNextTask който връща следващата незаета задача.

–Метод isAllWorkDone, който проверява, дали всички задачи са направени (една задача е направена, ако по нея остават 0 часа работа)

Да се спазва принципа за капсулация на данни!!!

В класът Employee (освен кода от преди), да се добави:



–поле allWork – референция към множеството от всичките задачи.

Всички работници работят с едно множество от задачи

Всеки работник разполага всеки ден с 8 часа за работа.

Целта е да се направи симулация на работен процес в офис, където има много работници които целят да свършат всичката работа (AllWork).

Всеки един работник работи всеки ден, докато не му свърши времето за деня, или докато всички задачи не са направени. Ако работник свърши

цялата работа по текущата му задача, той взема следващата свободна задача (метода `getNextTask`) и продължава да работи.

Симулацията ще се разиграва на дни, като деня ще свършва когато всички работници си изработят часовете. Тогава ще започва и нов ден.

–Направете метод на класа `Employee` `startWorkingDay()` който просто задава нови 8 часа за работа на работник (това ще става в началото на всеки ден)

–Променете метода `work()`, така, че да отговаря на условията описани по-горе: във всеки един момент всеки работник има 1 текуща задача (ако няма такава, си взема следващата свободна). Той работи по нея докато не му свърши времето за деня (края на деня). На другия ден продължава работа по нея. И така докато не завърши задачата. След това взема следващата свободна задача. Възможно е да свърши новата задача в рамките на същият ден, тогава отново си тегли задача. И така това се повтаря, ден след ден, докато не свършат свободните задачи.

Да се създаде програма в която се създава обект от клас `AllWork`, който се запълва със задачи(10-12 задачи) с различно време. Да се създадът и няколко (3-4) работника. В един безкраен цикъл, да се даде старт на работата и всички работници да започнат да си теглят задачи и да работят по тях. Всяка итерация на цикъла е 1 ден в офиса и приключва когато всичките работници си изработят 8те часа за деня.

Методите в класа `Employee` да се допълнят с подходящи съобщения, така че да се извеждат съобщения при всяко действие от страна на някой работник (взимане на нова задача, работене по текущата задача), за начало на нов работен ден и всичко необходимо за да се проследи изпълнението на програмата.

След като се стартира програмата, да започнат да текът дните, всички работници трябва да започват да работят и да се извежда съобщение за това в кой ден, кой работник какво да прави.