

Лекция 6

Референции, конструктори, методи и ключова дума return

Задача за домашно:

Задача 1: Допълнете задачата от предишното домашно:

Класът Computer да има следните конструктори:

-конструктор по подразбиране `Computer()`, който инициализира полето `isNotebook` със стойност `false`, и полето `operationSystem` с „Win XP“.

-конструктор с параметри `Computer(int year, double price, double hardDiskMemory, double freeMemory)`, който първо извиква конструктора по подразбиране и после инициализира останалите полета със стойностите подадени като аргументи.

-конструктор с параметри `Computer(int year, double price, boolean isNotebook, double hardDiskMemory, double freeMemory, String operationSystem)`, който инициализира всички полета със стойностите подадени като аргументи на конструктора.

-метод `int comparePrice(Computer c)`, 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.

В рамките на `main` метода да се създадат няколко обекта от тип `Computer` (чрез някой от наличните конструктори).

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

Задача 2: Създайте клас `Student`, описващ студент
Класът да има следните полета:

`name` – име на студента
`subject` – специалност
`grade` – успех
`yearInCollege` – курс
`age` – години на студента

`isDegree` – дали е завършил или не
`money` – пари от стипендии

Класът да дефинира следните конструктори:

-конструктор по подразбиране, който задава следните стойности на полетата: `grade = 4.0`, `yearInCollege = 1`, `isDegree = false`, `money = 0`

-конструктор с параметри, който извиква конструктора по подразбиране и инициализира(задава първоначални стойности) останалите полета на класа

Класът да дефинира следните методи:

-метод `void upYear()`, който увеличава годината в колежа с единица, ако студента не е завършил, в противен случай извежда подходящо съобщение. Ако след увеличаването, годината стане 4, задава `true` на полето `isDegree`

-метод `double receiveScholarship(double min, double 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 клас `StudentGroup`, репрезентиращ група от студенти от една и съща специалност.

Полета на класа:

`groupSubject` – специалност на студентите в групата

`Student [] students` – Студенти в групата (масив от обекти от клас `Student`)

`freePlaces` – свободни места в групата

Класът да дефинира следните конструктори:

-конструктор по подразбиране, който създава места за 5 студента в групата (инициализира полето `students` с нов масив от 5 елемента) и задава стойност 5 на полето `freePlaces`

-конструктор `StudentGroup(String subject)`, който първо извиква конструктора по подразбиране и задава стойност на полето `groupSubject`

Класът да дефинира следните методи:



-метод `void addStudent(Student s)`, който добавя студент към групата (добавя го към масива на следващата свободна позиция) 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 `void emptyGroup()`, който освобождава всички места в групата (задава нов празен масив за полето `students`) и задава стойност 5 за полето `freePlaces`.

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

-метод `printStudentsInGroup()` - Изкарва информация (Име, успех, ...) за всички студент в курса.

Да се напише клас `College` (с `main` метод), който демонстрира използването на класовете `Student` и `StudentGroup`

(Създава няколко студента, добавя стипендия на някои от тях, прехвърля някои от тях в по-горна година. Създава няколко групи от студенти, добавя студенти в тях, изкарва името на най-добрия студент от някоя от групите....)