

Контролно No. 2

Приложно Обектно Ориентирано Програмиране част 1

Инструкции:

1. Решете всички задачи. **Време за изпълнение 90 минути**
2. Да се качи в **Moodle** със **студентския акаунт** архивирано копие на целия проект на IntelliJ, именувано с **факултетния номер** на студента
2. **Използвайте дадените означения** за класове, променливи и методи.

Скала за оценяване:

- | | |
|---|--------------------|
| 2 | от 0 до 54 точки |
| 3 | от 55 до 64 точки |
| 4 | от 65 до 74 точки |
| 5 | от 75 до 84 точки |
| 6 | от 85 до 100 точки |

Решете следните задачи като спазите изискванията за **капсулиране, скриване на информация и повторно използване на код.**

Забележка: При установено **преписване** се пише **0 точки** за контролното

Задание за програмиране

A. Създайте **Java модулно приложение** в IntelliJ като го именувате с Вашия факултетен номер. Нека първоначално приложението да не съдържа други модули.

Добавете в така създаденото приложение `module` и съответен на него `package`, именувани `fnumber.lab`, където вместо `fnumber` изпишете Вашият факултетен номер, примерно f12345 Напишете следните класове в пакета `fnumber.lab` (общо 6 точки)

1. **Напишете** `enum Brand` в пакета `fnumber.lab`, който има следните елементи `HP`, `DELL`, `ASUS`, `LENOVO` (общо 6 точки)
2. **Напишете** `class Computer` в пакета `fnumber.lab`. Обектите на този клас имат следните данни: (общо 20 точки)

- Нека всеки обект от `class Computer` има уникален фабричен код `id(String` от 10 цифри с водещи незначещи нули, константа), `make` (име на производител от тип `Brand`), `price` (цена от тип `double`) и `registrationDate` (дата на закупуване от тип `LocalDate`).
- Напишете `GET` методи за `id`, `make`, `price` и `registrationDate`. Напишете `SET` методи за `make`, `price` и `registrationDate`, които да валидират по подходящ начин тези клас данни, съобразен с контекста на задачата
- Напишете конструктори- за общо ползване, по подразбиране и копиране. По подразбиране, името на производителя е `Brand.HP`, цената е 500.00, а датата на закупуване е днешната дата, `LocalDate.now()`
- Нека да има и `toString()` метод за извеждане на текущите стойности за клас данните, всяка на отделен ред със съответен промпт (**обяснителен текст**)
`ID. <id>: <make> , <price> , < registrationDate >`.

3. Напишете `class ComputerLab` в пакета `fnumber.lab`, който описва компютрите в една компютърна зала. Обектите на този клас имат следните данни: -(**общо 20 точки**)
✓ `computers` който е едномерен масив с елементи от тип `Computer`

✓ *LAB_ID*, който е уникална (неповтаряща се) константа от тип *String*, задаваща наименованието на залата и имаща вида *Lab001*

- Напишете SET и GET методи за *computers*, както и GET метод за *LAB_ID* на обектите от този клас
- Напишете трите вида конструктори за този клас – конструктор за общо ползване, конструктор по подразбиране (*computers* е масив с 30 елемента), и конструктор за копиране
- Напишете *String toString()* метод, връщащ форматиран текст с всички данни на текущия обект от този клас, по следния начин

Зала: < наименование >

<преминаване на нов ред>

(за всеки обект от тип Computer в computers се изписва на нов ред)

ID. <id>: <make> , <price> , < registrationDate >

Когато *computers* не съдържа обекти от тип *Computer* да се изведе подходящ текст, който информира за това.

4. Напишете в *class ComputerLab* (общо 10 точки)

✓ Напишете метод

public int[] groupByMonth(int byYear) , който връща масив с броя на компютрите регистриране през месеците януари, февруари,..., ноември, декември в годината *byYear*. При липса на данни за *byYear*, методът връща *null*.

5. Добавете подходящо дефиниран *module-info.java* за модул *fnumber.lab*, който да позволява коректно компилиране на цялото приложение. Създайте JAR файл с *Brand*, *Computer* и *ComputerLab* в модул *fnumber.lab*. (общо 6 точки)

Б. Добавете *module* и съответен на него *package*, именувани *fnumber.test*. Създайте *class ComputerLabTest* в *package fnumber.test* на този модул. (общо 6 точки)---32

1. Напишете *public static void main(String... args) method* в *class ComputerLabTest* за тестване на *class ComputerLab* (общо 24 точки)

- а) Създайте масив *hardware* с 20 елемента от тип *Computer* като използвате методите на обект от клас *Random* на *make* (произволно избран елемент от *Brand*) , *price* (произволно избрана стойност в интервала [500, 1500]) и *registrationDate* (ден 1, месец произволно избран в интервала [1,12] и година 2020) (6 точки)
- б) Създайте обект *lab* от *class ComputerLab* с така създадения масив *hardware* (2 точки)
- в) Изведете с *toString()* данните на обекта *lab* (2 точки)
- г) Изведете на стандартен изход отговор на въпроса „дали в *lab* средният брой на компютрите, регистрирани през първото тримесечие е по- голям от средния брой компютри, регистрирани по месеци през цялата година?“ Среден брой на нула на брой компютри е нула. (10 точки)

е) Създайте обект *newLab* от *class ComputerLab* по подразбиране (2 точки)

ф) Изведете с *toString()* данните на обекта *newLab* (2 точки)

2. Добавете подходящо дефиниран *module-info.java* за модул *fnumber.test*, който да позволява коректно компилиране на цялото приложение. Създайте и конфигурирайте коректно *Application template* за изпълнение на цялото приложение. (общо 6 точки)