

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.data`, където вместо `fnumber` изпишете Вашият факултетен номер, примерно f12345. Напишете следните класове в пакета `fnumber.data` (**общо 6 точки**)

1. Напишете `class Car`, който описва лека кола, в пакета `fnumber.data`. Обектите на този клас имат следните данни: (**общо 20 точки**)

- a) `NOMER` от тип `String`, който е уникална (неповтаряща се) константа, представляваща номера на колата от вида "СВ `xxxx`" (`xxxx` е четирицифрено число, където незначещите цифри са заменени с нули).
 - b) `model` от тип `String`, представя наименованието на модел лека кола
 - c) `abonament` от тип `int`, брой месеци [1, 12], за които е платено колата да бъде на паркинг
- Напишете `SET` и `GET` методи за данните `model` и `abonament`, както и `GET` метод за `NOMER` на обектите от този клас
 - Напишете трите вида конструктори за този клас – конструктор за общо ползване, конструктор по подразбиране (абонаментът е 1 месец,, моделът е „Неизвестен“) и конструктор за копиране
 - Напишете `String toString()` метод, връщащ форматиран текст с данните на текущия обект от този клас, по следния начин
`Кола:<наименование на модела>, Номер:<номер на колата> <нов ред>`
`Брой платени месеци: <абонамент>`

2. Напишете `class Parking` в пакета `fnumber.data`, който описва колите на даден паркинг. Обектите на този клас имат следните данни: -(**общо 21 точки**)

- a) *TOTAL_CARS* е *public* константа от тип *int*, задаваща максималният брой коли на всеки паркинг (обект от *class Parking*). Нека *TOTAL_CARS* да е 50.
- b) *cars* от тип *Car []*, който е масив от *TOTAL_CARS* на брой леки коли, всеки от които е обект от тип *Car*.
- c) *location* от тип *String*, който е адреса на паркинга
- Напишете SET и GET методи за данните *cars* и *location* на обектите от този клас
- Напишете трите вида конструктори за този клас – конструктор за общо ползване, конструктор по подразбиране (*cars* е масив с *TOTAL_CARS* елемента, *location* е „unknown“), и конструктор за копиране
- Напишете *String toString()* метод, връщащ форматиран текст с всички данни на текущия обект от този клас, по следния начин
Address: < location >, Total cars: < TOTAL_CARS >, <преминаване на нов ред>
(за всеки обектите от тип Car, добавени в масива cars , се създава текст)
car ID <номер на кола> <TAB> Subscription: < abonament >
<преминаване на нов ред>

3. В *class Parking* -(общо 11 точки)

- Напишете метод
public int[] paymentByMonth() , който връща броя на колите в паркинга (масива *cars*) с платен абонамент за 1, 2, 3, 4 ..., 12 месеца.

4. Добавете подходящо дефиниран module-info.java за модул *fnumber.data*, който да позволява коректно компилиране на цялото приложение. Създайте JAR файл с класовете *Car* и *Parking* в модул *fnumber.data*. (общо 6 точки)

Б. Добавете module и съответен на него package, именуван *fnumber.test*. Създайте *class ParkingTest* в package *fnumber.test* на този модул. (общо 6 точки)

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

- a) създайте масив *cars* от *Parking.TOTAL_CARS* елемента от тип *Car*. Запълнете 60% от елементите на масив *cars* с елементи, които се инициализират по случаен начин с помощта на обект от *class Random* по следния начин: (10 точки)
 - абонаментът се избира по случаен начин в интервала [1– 12]
 - моделът на колата е с наименование “Opel” или “Volvo” в зависимост от това дали генераторът е избрал случайно съответно 0 или 1
- b) създайте обект *parking* от тип *Parking* с конструктор за общо ползване като използвате адрес на паркинга по Ваш избор и масива *cars* , създаден в предходната точка (a) (2 точки)

- c) Изведете на стандартен изход обекта *parking* като използвате метода *toString()* на клас *Parking* (2 точки)
- d) Изведете на стандартен изход отговора на въпроса дали средният брой на коли в паркинга с абонамент за 1, 4 и 8 месеца е по-голям от средния брой на колите с абонамент за 5 и 9 месеца. Среден брой на нула на брой абонамента е нула. (10 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)