

Упражнение: Преглед на MVC

Решете следните познати Ви вече задачи от курса „Увод в програмирането“, като модификацията в условието е да се създаде конзолно MVC приложение.

Подсказка

Разгледайте решените вече задачи и модифицирайте кода, така, че да имате:

- Главна програма с клас Program, чиято единствена задача ще е да създаде един обект от клас **SomethingActionController()** с оператор **new**

```
SomethingActionController something= new SomethingActionController(); ,
```

Където **Something** може да бъде заменено с името на класа от оригиналната задача. Т.е. по смисъла на данните, които се обработват в конкретната задача.

- Клас **SomethingActionController()**, който управлява обектите **Something** и **Display** за да може приложението да въвежда и извежда необходимите данни.
- Клас **Something**, в който да съхранявате необходимите за конкретната задача данни
- Клас **Display**, който се грижи да въвежда и извежда данни, които да предава на класа **Something**

Ето и задачите:

Задача 1. Цена за транспорт

Студент трябва да пропътува **n** километра. Той има избор измежду **три вида транспорт**:

- Такси.** Начална такса: **0.70** лв. Дневна тарифа: **0.79** лв. / км. Нощна тарифа: **0.90** лв. / км.
- Автобус.** Дневна / нощна тарифа: **0.09** лв. / км. Може да се използва за разстояния минимум **20** км.
- Влак.** Дневна / нощна тарифа: **0.06** лв. / км. Може да се използва за разстояния минимум **100** км.

Напишете програма, която въвежда броя километри **n** и период от деня (ден или нощ) и изчислява **цената на най-евтиния транспорт**.

Вход

От конзолата се четат **два реда**:

- Първият ред съдържа числото **n** – брой километри – цяло число в интервала [1...5000].
- Вторият ред съдържа дума **“day”** или **“night”** – пътуване през деня или през нощта.

Изход

Да се отпечата на конзолата **най-ниската цена** за посочения брой километри.

Примерен вход и изход

Вход	Изход	Обяснения
5 day	4.65	Разстоянието е под 20 км → може да се ползва само такси . Началната такса е 0.70 лв. Понеже е през деня, тарифата е 0.79 лв. / км. С такси цената е: $0.70 + 5 * 0.79 = 4.65$ лв.

7 night	7	Разстоянието е под 20 км → може да се ползва само такси . Началната такса е 0.70 лв. Понеже е през нощта, тарифата е 0.90 лв. / км. С такси цената е: $0.70 + 7 * 0.90 = 7.00$ лв.
25 day	2.25	Разстоянието е над 20 км → може да се ползва автобус , но не може да се ползва влак. Автобусът е най-евтиния възможен вариант. С автобус цената е: $25 * 0.09 = 2.25$ лв.
180 night	10.8	Разстоянието е над 100 км → може да се ползва влак . Влакът е най-евтиният възможен вариант за пътуване. С влак цената е: $180 * 0.06 = 10.80$ лв.

Задача 2. Навреме за изпит

Студент трябва да отиде **на изпит в определен час** (например в 9:30 часа). Той идва в изпитната зала в даден **час на пристигане** (например 9:40). Счита се, че студентът е дошъл **навреме**, ако е пристигнал в часа на изпита или до половин час преди това. Ако е пристигнал по-рано повече от 30 минути, той е **подранил**. Ако е дошъл след часа на изпита, той е **закъснял**. Напишете програма, която въвежда време на изпит и време на пристигане и отпечатва дали студентът е дошъл **навреме**, дали е **подранил** или е **закъснял** и с **колко часа или минути** е подранил или закъснял.

Вход

От конзолата се четат **4 цели числа** (по едно на ред):

- Първият ред съдържа **час на изпита** – цяло число от 0 до 23.
- Вторият ред съдържа **минута на изпита** – цяло число от 0 до 59.
- Третият ред съдържа **час на пристигане** – цяло число от 0 до 23.
- Четвъртият ред съдържа **минута на пристигане** – цяло число от 0 до 59.

Изход

На първият ред отпечатайте:

- **"Late"**, ако студентът пристига по-късно от часа на изпита.
- **"On time"**, ако студентът пристига точно в часа на изпита или до 30 минути по-рано.
- **"Early"**, ако студентът пристига повече от 30 минути преди часа на изпита.

Ако студентът пристига с поне минута разлика от часа на изпита, отпечатайте на следващия ред:

- **"mm minutes before the start"** за идване по-рано с по-малко от час.
- **"hh:mm hours before the start"** за подраняване с 1 час или повече. Минутите винаги печатайте с 2 цифри, например "1:05".
- **"mm minutes after the start"** за закъснение под час.
- **"hh:mm hours after the start"** за закъснение от 1 час или повече. Минутите винаги печатайте с 2 цифри, например "1:03".

Примерен вход и изход

Вход	Изход
9 30 9 50	Late 20 minutes after the start
9 00 8	On time 30 minutes before the start

Вход	Изход
9 00 10 30	Late 1:30 hours after the start
14 00 13	On time 5 minutes before the start

Вход	Изход
10 00 10 00	On time
11 30 10	Early 35 minutes before the start

30		55		55	
16 00 15 00	Early 1:00 hours before the start	11 30 8 12	Early 3:18 hours before the start	11 30 12 29	Late 59 minutes after the start

Задача 3. Хистограма

Дадени са **n** цели числа в интервала [1...1000]. От тях някакъв процент **p1** са под 200, друг процент **p2** са от 200 до 399, друг процент **p3** са от 400 до 599, друг процент **p4** са от 600 до 799 и останалите **p5** процента са от 800 нагоре. Да се напише програма, която изчислява и отпечатва процентите **p1**, **p2**, **p3**, **p4** и **p5**.

Пример: имаме **n = 20** числа: 53, 7, 56, 180, 450, 920, 12, 7, 150, 250, 680, 2, 600, 200, 800, 799, 199, 46, 128, 65. Получаваме следното разпределение и визуализация:

Диапазон	Числа в диапазона	Брой числа	Процент
< 200	53, 7, 56, 180, 12, 7, 150, 2, 199, 46, 128, 65	12	$p1 = 12 / 20 * 100 = 60.00\%$
200 ... 399	250, 200	2	$p2 = 2 / 20 * 100 = 10.00\%$
400 ... 599	450	1	$p3 = 1 / 20 * 100 = 5.00\%$
600 ... 799	680, 600, 799	3	$p4 = 3 / 20 * 100 = 15.00\%$
≥ 800	920, 800	2	$p5 = 2 / 20 * 100 = 10.00\%$

Вход

На първия ред от входа стои цялото число **n** ($1 \leq n \leq 1000$) – брой числа. На следващите **n** реда стои по едно цяло число в интервала [1...1000] – числата върху които да бъде изчислена хистограмата.

Изход

Да се отпечата на конзолата хистограмата – **5 реда**, всеки от които съдържа число между 0% и 100%, с точност две цифри след десетичната точка, например 25.00%, 66.67%, 57.14%.

Примерен вход и изход

Вход	Изход	Вход	Изход	Вход	Изход	Вход	Изход	Вход	Изход
3	66.67%	4	75.00%	7	14.29%	9	33.33%	14	57.14%
1	0.00%	53	0.00%	800	28.57%	367	33.33%	53	14.29%
2	0.00%	7	0.00%	801	14.29%	99	11.11%	7	7.14%
999	0.00%	56	0.00%	250	14.29%	200	11.11%	56	14.29%
	33.33%	999	25.00%	199	28.57%	799	11.11%	180	7.14%
				399		999		450	
				599		333		920	
				799		555		12	
						111		7	
						9		150	
								250	
								680	
								2	
								600	

								200	
--	--	--	--	--	--	--	--	-----	--

Задача 4. Генератор за тъпи пароли

Да се напише програма, която въвежда две цели числа n и L и генерира по азбучен ред всички възможни “тъпи” пароли, които се състоят от следните 5 символа:

- Символ 1: цифра от 1 до n .
- Символ 2: цифра от 1 до n .
- Символ 3: малка буква измежду първите L букви на латинската азбука.
- Символ 4: малка буква измежду първите L букви на латинската азбука.
- Символ 5: цифра от 1 до n , по-голяма от първите 2 цифри.

Вход

Входът се чете от конзолата и се състои от две цели числа n и L в интервала $[1...9]$, по едно на ред.

Изход

На конзолата трябва да се отпечатаат всички “тъпи” пароли по азбучен ред, разделени с интервал.

Примерен вход и изход

вход	изход
2 4	11aa2 11ab2 11ac2 11ad2 11ba2 11bb2 11bc2 11bd2 11ca2 11cb2 11cc2 11cd2 11da2 11db2 11dc2 11dd2
3 1	11aa2 11aa3 12aa3 21aa3 22aa3
3 2	11aa2 11aa3 11ab2 11ab3 11ba2 11ba3 11bb2 11bb3 12aa3 12ab3 12ba3 12bb3 21aa3 21ab3 21ba3 21bb3 22aa3 22ab3 22ba3 22bb3
4 2	11aa2 11aa3 11aa4 11ab2 11ab3 11ab4 11ba2 11ba3 11ba4 11bb2 11bb3 11bb4 12aa3 12aa4 12ab3 12ab4 12ba3 12ba4 12bb3 12bb4 13aa4 13ab4 13ba4 13bb4 21aa3 21aa4 21ab3 21ab4 21ba3 21ba4 21bb3 21bb4 22aa3 22aa4 22ab3 22ab4 22ba3 22ba4 22bb3 22bb4 23aa4 23ab4 23ba4 23bb4 31aa4 31ab4 31ba4 31bb4 32aa4 32ab4 32ba4 32bb4 33aa4 33ab4 33ba4 33bb4

Задача 5. Зеленчукова борса

Градинар продавал реколтата от градината си на зеленчуковата борса. Продава зеленчуци за N лева на килограм и плодове за M лева за килограм. Напишете програма, която да пресмята приходите от реколтата в евро (ако приемем, че едно евро е равно на 1.94лв).

Вход

От конзолата се четат 4 числа, по едно на ред:

- Първи ред – Цена за килограм зеленчуци – число с плаваща запетая
- Втори ред – Цена за килограм плодове – число с плаваща запетая
- Трети ред – Общо килограми на зеленчуците – цяло число
- Четвърти ред – Общо килограми на плодовете – цяло число

Ограничения: Всички числа ще са в интервала от 0.00 до 1000.00

Изход

Да се отпечата на конзолата **едно число с плаваща запетая**: приходите от всички плодове и зеленчуци в евро.

Примерен вход и изход

Вход	Изход	Обяснения
0.194 19.4 10 10	101	Зеленчуците струват – 0.194лв. * 10кг. = 1.94лв. Плодовете струват – 19.4лв. * 10кг. = 194лв. Общо – 195.94лв. = 101евро
1.5 2.5 10 10	20.6185567010309	

Задача 6. Тръби в басейн

Басейн с **обем V** има **две тръби** от които се пълни. **Всяка тръба има определен дебит** (литрите вода минаващи през една тръба за един час). Работникът **пуска тръбите едновременно** и излиза за **N** часа. Напишете програма, която изкарва състоянието на басейна, **в момента, когато работникът се върне**.

Вход

От конзолата се четат **четири реда**:

- Първият ред съдържа числото **V** – **Обем на басейна в литри** – цяло число в интервала [1...10000].
- Вторият ред съдържа числото **P1** – **дебит на първата тръба за час** – цяло число в интервала [1...5000].
- Третият ред съдържа числото **P2** – **дебит на втората тръба за час** – цяло число в интервала [1...5000].
- Четвъртият ред съдържа числото **N** – **часовете които работникът отсъства** – число с плаваща запетая в интервала [1.0...24.00]

Изход

Да се отпечата на конзолата **едно от двете възможни състояния**:

- До колко се е запълнил басейна и коя тръба с колко процента е допринесла. Всички проценти се свеждат до цяло число (без закръгляне).
 - "The pool is [x]% full. Pipe 1: [y]%. Pipe 2: [z]%. "
- Ако басейнът се е препълнил – с колко литра е прелял за даденото време, число с плаваща запетая
 - "For [x] hours the pool overflows with [y] liters. "

* **Имайте предвид**, че поради свеждането до цяло число се губят данни и нормално **сборът на процентите да е 99%, а не 100%**.

Примерен вход и изход

Вход	Изход	Обяснения
------	-------	-----------

1000 100 120 3	The pool is 66% full. Pipe 1: 45%. Pipe 2: 54%.	За 3 часа: Първата тръба е напълнила – 300 л. Втората тръба е напълнила – 360 л. Общо – 660 л. < 1000 л. => 66% са запълнени Първата тръба е допринесла с 45% (300 от 660 л.). Втората тръба е допринесла с 54% (360 от 660 л.).
100 100 100 2.5	For 2.5 hours the pool overflows with 400 liters.	За 2.5 часа: Първата тръба е напълнила – 250 л. Втората тръба е напълнила – 250 л. Общо – 500 л. > 100 л. => 400 л. са преляли.

Задача 7. Пътешествие

Странно, но повечето хора си плануват от рано почивката. Млад програмист разполага с **определен бюджет** и свободно време в даден **сезон**. Напишете програма, която да приема **на входа бюджета и сезона**, а **на изхода** да изкарва, **къде ще почива** програмиста и **колко ще похарчи**.

Бюджета определя дестинацията, а сезона определя колко от бюджета ще изхарчи. Ако е **лято** ще почива на **къмпинг**, а **зимата** в **хотел**. Ако е в **Европа**, **независимо от сезона** ще почива в **хотел**. Всеки **къмпинг** или **хотел**, **според дестинацията**, има **собствена цена** която отговаря на даден **процент от бюджета**:

- При **100лв. или по-малко** – някъде в **България**
 - **Лято** – **30%** от бюджета
 - **Зима** – **70%** от бюджета
- При **1000лв. или по малко** – някъде на **Балканите**
 - **Лято** – **40%** от бюджета
 - **Зима** – **80%** от бюджета
- При **повече от 1000лв.** – някъде из **Европа**
 - При пътуване из Европа, независимо от сезона ще похарчи **90% от бюджета**.

Вход

Входът се чете от конзолата и се състои от **два реда**:

- **Първи ред** – Бюджет, **реално число** в интервала **[10.00...5000.00]**.
- **Втори ред** – Един от двата възможни сезона: „summer” или „winter”

Изход

На конзолата трябва да се отпечата **два реда**.

- **Първи ред** – „Somewhere in [дестинация]” измежду „Bulgaria”, „Balkans” и „Europe”
- **Втори ред** – “{Вид почивка} – {Похарчена сума}”
 - Почивката може да е между „Camp” и „Hotel”
 - Сумата трябва да е **закръглена с точност до вторият знак след запетаята**.

Примерен вход и изход

вход	изход
50	Somewhere in Bulgaria

summer	Camp - 15.00
75 winter	Somewhere in Bulgaria Hotel - 52.50
312 summer	Somewhere in Balkans Camp - 124.80
678.53 winter	Somewhere in Balkans Hotel - 542.82
1500 summer	Somewhere in Europe Hotel - 1350.00

Задача 8. Деление без остатък

Дадени са **n цели числа** в интервала [1...1000]. От тях някакъв **процент p1 се делят без остатък на 2**, друг **процент p2 се делят без остатък на 3**, друг **процент p3 се делят без остатък на 4**. Да се напише програма, която изчислява и отпечатва процентите **p1, p2 и p3**.

Пример: имаме **n = 10** числа: 680, 2, 600, 200, 800, 799, 199, 46, 128, 65. Получаваме следното разпределение и визуализация:

Деление без остатък на:	Числа в диапазона	Брой числа	Процент
2	680, 2, 600, 200, 800, 46, 128	7	$p1 = 7.0 / 10 * 100 = 70.00\%$
3	600	1	$p2 = 1 / 10 * 100 = 10.00\%$
4	680, 600, 200, 800, 128	5	$p3 = 5 / 10 * 100 = 50.00\%$

Вход

На първия ред от входа стои цялото число **n** ($1 \leq n \leq 1000$) – брой числа. На следващите **n реда** стои **по едно цяло число** в интервала [1...1000] – числата които да бъдат проверени на колко се делят.

Изход

Да се отпечата на конзолата **3 реда**, всеки от които съдържа процент между 0% и 100%, с точност две цифри след десетичната точка, например 25.00%, 66.67%, 57.14%.

- На **първият ред** – процентът на числата които **се делят на 2**
- На **вторият ред** – процентът на числата които **се делят на 3**
- На **третият ред** – процентът на числата които **се делят на 4**

Примерен вход и изход

Вход	Изход	Вход	Изход
10	70.00%	3	33.33%
680	10.00%	3	100.00%
2	50.00%	6	0.00%
600		9	
200			
800			
799			

199				
46				
128				
65				

Задача 9. Магически числа

Да се напише програма, която въвежда едно цяло „магическо“ число и изкарва **всички** възможни **6-цифрени числа**, за които **произведението на неговите цифри** е **равно** на „магическото“ число.

Пример: „Магическо число“ -> 2

- 111112 -> $1 * 1 * 1 * 1 * 1 * 2 = 2$
- 111121 -> $1 * 1 * 1 * 1 * 2 * 1 = 2$
- 111211 -> $1 * 1 * 1 * 2 * 1 * 1 = 2$
- 112111 -> $1 * 1 * 2 * 1 * 1 * 1 = 2$
- 121111 -> $1 * 2 * 1 * 1 * 1 * 1 = 2$
- 211111 -> $2 * 1 * 1 * 1 * 1 * 1 = 2$

Вход

Входът се чете от конзолата и се състои от **едно цяло число** в интервала [1...600000].

Изход

На конзолата трябва да се отпечата **всички** „магически“ числа, разделени с **интервал**.

Примерен вход и изход

вход	изход
2	111112 111121 111211 112111 121111 211111
8	111118 111124 111142 111181 111214 111222 111241 111412 111421 111811 112114 112122 112141 112212 112221 112411 114112 114121 114211 118111 121114 121122 121141 121212 121221 121411 122112 122121 122211 124111 141112 141121 141211 142111 181111 211114 211122 211141 211212 211221 211411 212112 212121 212211 214111 221112 221121 221211 222111 241111 411112 411121 411211 412111 421111 811111
531441	999999

Задача 10. Ремонт на плочки

На **площадката** пред жилищен блок трябва да се **поставят плочки**. Площадката е с форма на **квадрат** със **страна N метра**. Плочките са **широки „W“ метра** и **дълги „L“ метра**. На площадката има **една пейка** с **ширина M метра** и **дължина O метра**. Под нея **не е нужно** да се **слагат плочки**. **Всяка плочка** се поставя за **0.2 минути**.

Напишете **програма**, която **чете от конзолата размерите** на **площадката**, **плочките** и **пейката** и **пресмята колко плочки са необходими** да се **покрие площадката** и **пресмята времето за поставяне на всички плочки**.

Пример: **площадка** с **размер 20м.** има **площ 400кв.м.** **Пейка** **широка 1м.** и **дълга 2м.**, **заема площ 2кв.м.** **Една плочка** е **широка 5м.** и **дълга 4м.** има **площ = 20кв.м.** **Площта** която трябва да се **покрие** е **400 – 2 = 398 кв.м.** **Необходими са 398 / 20 = 19.90 плочки.** **Необходимото време** е **19.90 * 0.2 = 3.98 минути.**

Вход

От конзолата се четат **5** числа:

- **N** – дължината на страна от площадката в интервала **[1...100]**
- **W** – широчината на една плочка в интервала **[0.1...10.00]**
- **L** – дължината на една плочка в интервала **[0.1...10.00]**
- **M** – широчината на пейката в интервала **[0...10]**
- **O** – дължината на пейката в интервала **[0...10]**

Изход

Да се отпечата на конзолата две числа: **броят плочки** необходими за ремонта и **времето за поставяне**, всяко на нов ред.

Примерен вход и изход

Вход	Изход	Обяснения
20 5 4 1 2	19.9 3.98	Обща площ = $20 * 20 = 400$; площ на пейката = $1 * 2 = 2$ Площ за покриване = $400 - 2 = 398$ Площ на плочки = $5 * 4 = 20$ Необходими плочки = $398 / 20 = 19.9$ Необходимо време = $19.9 * 0.2 = 3.98$
40 0.8 0.6 3 5	3302.083333333333 660.4166666666667	

Министерство на образованието и науката (МОН)

- Настоящият курс (презентации, примери, задачи, упражнения и др.) е разработен за нуждите на Национална програма **"Обучение за ИТ кариера"** на МОН за подготовка по професия "Приложен програмист".



Министерство
на образованието
и науката



- Курсът е базиран на учебно съдържание и методика, предоставени от **фондация "Софтуерен университет"** и се разпространява под **свободен лиценз CC-BY-NC-SA** (Creative Commons Attribution-Non-Commercial-Share-Alike 4.0 International).

