

Проекти за завършване на ниво Java101 - 2023 г

1. Bar Me!

Програмата трябва да съдържа данни за поне 10 от популярните заведения в Разград / Търговище - име, географска локация, работно време. При стартиране на програмата, потребителят трябва да въведе текущата си локация (дължина и ширина). След това се появява меню с три опции.

Първата опция е СПИСЪК ВСИЧКИ. Ако потребителят избере нея, на екрана се появява списък с баровете, подредени от най-близък до най-далечен до потребителя. Всеки ред от списъка съдържа пореден номер, името на бара и разстояние в метри до потребителя, както и работното време.

Втората опция е СПИСЪК ОТВОРЕНИ. На екрана се появяват само баровете, които в момента са отворени. Най-отгоре са тези, които затварят най-скоро. Всеки ред от списъка съдържа пореден номер, името на бара и работното време.

Третата опция е КАРТА. На екрана се появява 1Д "карта" (линия), съставена от ASCII символи. С X е отбелязана локацията на потребителя, с цифрите от 1 до броя барове са отбелязани баровете. Под картата се изписва и списък, показващ коя цифра на кой бар отговаря. Всеки символ от картата отговаря на 50м от реалния свят.

Пример:

```
>> Въведете локацията си
<< 250м
>> Изберете опция: СПИСЪК ВСИЧКИ (1), СПИСЪК ОТВОРЕНИ (2), КАРТА (3)
<< 1
>> 1. Famous (21.00 - 3.00) - 150м
    2. Soho (8.00 - 24.00) - 400м
    3. Милениум (21.00 - 3.00) - 750м
    4. ....
```

Изберете опция: СПИСЪК ВСИЧКИ (1), СПИСЪК ОТВОРЕНИ (2), КАРТА (3)

<< 3

___1_X___2___3

1. Famous
2. Soho
3. Милениум

2. Детектор за плагиатство

Authorship detection е процесът на алгоритмично идентифициране на автора на даден анонимен текст. Основната идея е извличане на различни статистики от текста (т.нар. feature-и в контекста на Machine Learning) с цел формиране на лингвистичен подпис (linguistic signature) за всеки текст.

Един пример за feature е средният брой думи в изречение.

Имайки лингвистичните подписи на два текста (т.е. поредици от числа, всяко от които съответства на стойността на даден feature), можем да определим сходството между тях и да изчислим каква е вероятността да са написани от един и същ автор.

Автоматизираният authorship detection днес е област на активен изследователски интерес и има приложения в plagiarism detection, филтрирането на имейли, проучвания в социалните науки и дори в съдебната практика като доказателство в дела. Конкретната задача е вдъхновена от курс по програмиране във Факултета по компютърни науки на University of Toronto.

Ще използваме следните feature-и за идентифициране автора на даден текст:

1. Средна дължина на думите - средният брой символи в дума, след strip-ване на пунктуацията.
2. Type-Token Ratio - броят на всички различни думи, използвани в текста, разделен на броя на всички думи. Измерва колко повтаряща се е лексиката.
3. Нарах Legomena Ratio - броят на думите, срещащи се само по веднъж в даден текст, разделен на броя на всички думи.
4. Среден брой думи в изречение - броят на всички думи, използвани в текста, разделен на броя на изреченията.

Програмата приема текст, изкарва стойностите на feature-ите му, приема втори текст, също изкарва стойностите на feature-ите му и след това изкарва резултат на сходство между двата текста.

Резултата на сходство се пресмята по формулата: $\text{abs}(F1T1-F1T2) * 11 + \text{abs}(F2T1-F2T2) * 33 + \text{abs}(F3T1-F3T2) * 50 + \text{abs}(F4T1-F4T2) * 0.4$, където $F1T1$ е feature 1 (средна дължина на думите) за текст 1, $F1T2$ е feature 1 за текст 2,

F2T1 е feature 2 (type-token ratio) за текст 1 и т.н. Числата 11, 33, 50, 0.4 са тежести. Направете програмата така че да могат те лесно да се променят.

Пример:

>> Enter first text:

<< this is the

first sentence. Isn't it? Yes ! !! This

last bit :) is also a sentence, but
without a terminator other than the end of the file

>> 1. Avg. word length: 4
2. Type-Token Ratio: 3.333
3. Hapax Legomena Ratio: 0.538
4. Avg. sentence length: 6.5

>> Enter second text:

<< Hello! This is just a simple text. Carry on!

>> 1. Avg. word length: 3.666
2. Type-Token Ratio: 1
3. Hapax Legomena Ratio: 1
4. Avg. sentence length: 3

>> Similarity: $3.666 + 76.989 + 23.1 + 1.4 = 105.155$

3. Дати

Да се напише програма, която поддържа текстов диалогов режим, позволяващ удобен интерактивен избор на серия от операции с дати:

- въвеждане на дата от клавиатурата и проверка за коректност
- извеждане на дата на екрана в избран от потребителя формат
допустими формати: dd/MM/yyyy -> 20/04/2021
MM/dd/yyyy -> 04/20/2021
dd-MMM-yyyy -> 04-Feb-2021
- по дата да се определи ден от седмицата
- по описание с ден от седмицата да се определи датата, която отговаря на него, например:
 - втората събота от месец февруари
 - последният понеделник от месец януари
- отпечатва календар за даден месец
 - календарът да се извежда прегледно, като си личи коя дата на кой ден от седмицата отговаря

При стартиране на програмата потребителя въвежда кой ден от седмицата е 1 януари и спрямо този ден динамично се генерира календара до края на годината. Денят се въвежда като число от 1 до 7, като 1 отговаря на понеделник, а 7 на неделя. Предварително се знае, че годината е високосна.

Реализацията на всички функции трябва да се съдържа в дадена програма, не е позволено използването на системна библиотека за работа с дати.

Пример:

>>5

	П	В	С	Ч	П	С	Н
					1	2	3
ЯНУАРИ ->	4	5	6	7	8	9	10
	11	12	13	14	15	16	17
	18	19	20	21	22	23	24
	25	26	27	28	29	30	31

	П	В	С	Ч	П	С	Н
	1	2	3	4	5	6	7
ФЕВРУАРИ ->	8	9	10	11	12	13	14
	15	16	17	18	19	20	21
	22	23	24	25	26	27	28
	29						

>> втората

събота от месец февруари

<< 13/02

>> последният понеделник от месец януари

<< 28/01

4. Склад

Да се напише компютърна програма, реализираща информационна система, обслужваща склад. Програмата да поддържа текстов диалогов режим, позволяващ удобен интерактивен избор на следните операции:

- списък на наличните продукти в склада. За всеки продукт се съхранява и извежда следната информация:
 - име (описание - символен низ с произволна дължина)
 - срок на годност
 - дата на постъпване в склада
 - име на производител
 - мерна единица (килограми, литри)
 - налично количество
 - местоположение (секция/рафт/номер) (номерируйте склада си както прецените, че ще ви е удобно, имайте предвид, че в началото той е празен и различно количество стока е нормално да заема различно по обем място)
 - допустим брой от продукта на рафт
 - коментар (свободен текст)

- записване на нова доставка
 - ако нов продукт е с различен срок на годност от вече съществуващ едноименен продукт, той да бъде поставен на различно място
 - ако имате достатъчно място, еднакви продукти с един и същи срок на годност да бъдат поставени на едно и също място
 - при извеждане на списъка с налични продукти да се изведе общото количество на едноименните продукти независимо от срока им на годност
- справка за наличността в даден период (по дадена начална и крайна дата се извежда списък с всички промени на наличността в дадения период, включително зареждания и извеждания на стоки)

Пример:

Please choose what to do (1 - List all items; 2 - Add new delivery; 3 - List deliveries for time period)

<< 1

Light bulb - LED 75W | Expiry date: n/a | Entry date: 05.05.2021 | Manufacturer: Philips | Unit: Item | Stock: 104 | Position: A3 / 4 / 10 | Available items at shelf: 500 | Comment:

Battery AAA | Expiry date: 10.10.2026 | Entry date: 10.07.2020 | Manufacturer: Duracel | Unit: Item | Stock: 638 | Position: C2 / 5 / 1 | Available items at shelf: 10000 | Comment: in every box there are 10 packages of 4 batters each

Please choose what to do (1 - List all items; 2 - Add new delivery; 3 - List deliveries for time period)

<< 2

Enter product name:

<< Battery CR32

Enter expiry date:

<< 24.11.2025

Enter entry date:

<< 02.06.2021

Enter manufacturer:

<< Varta

Enter unit:

<< Item

Enter available stock:

<< 900

Enter comment (optional):

<<

Product was added successfully!

Please choose what to do (1 - List all items; 2 - Add new delivery; 3 - List deliveries for time period)

<< 3

From date:

<< 01.05.2021

To date:

<< 31.05.2021

List of product transactions for the period 01.05.2021 - 31.05.2021:

- Light bulb - LED 75W | Entry date: 05.05.2021 | Unit: Item | Stock delivered: 200

5. Личен календар

Да се напише компютърна програма, реализираща информационна система, която поддържа личен календар. Програмата да поддържа текстов диалогов режим, позволяващ удобен интерактивен избор на следните операции:

- запазване/отказване на час за среща. Записва се начало на среща (ден, час), край на среща, име (кратко описание - например "зъболекар") и коментар
- отпечатване на дневна програма, като по даден ден се извежда хронологичен списък с всички ангажменти за деня
- промяна на часа и/или мястото на среща по име, като се запази останалата информация за събитието
- търсене на среща: по име се извеждат останалите данни се срещата
- намиране на свободно място за среща: по дадена начална дата и желана продължителност на срещата търси дата, на която е възможно да се запази такава среща, но само в работни дни и не преди 8 часа сутринта или след 5 часа вечерта.

Пример:

Please choose what to do (1 - Create an event | 2 - Daily schedule | 3 - Search event | 4 - Find availability)

<< 1

Enter event name:

<< Java lecture

Enter event date:

<< 17.06.2021

Enter start time:

<< 18:00

Enter end time:

<< 20:00

Enter notes:

<< There will be project assignment at this lecture

Event was created succesfully!

Please choose what to do (1 - Create an event | 2 - Daily schedule | 3 - Search event | 4 - Find availability)

<< 2

Enter date:

<< 10.06.2021

- 8:00 - 9:00 Sport: running
- 10:30 - 11:00 Daily team meeting
- 12:00 - 13:00 Lunch break: at Pizza Milevi
- 16:00 - 17:00 Technical planning: planning the release of next features

Please choose what to do (1 - Create an event | 2 - Daily schedule | 3 - Search event | 4 - Find availability)

<< 3

Enter event title:

<<Dentist

25.05.2021 15:30 - 16:30 At the dentist: Fixing my teeth with caries

Do you want to Cancel or Edit event (1 - Cancel | 2 - Edit | press any other key to continue)

<< 1

Are you sure you want to cancel the event 25.05.2021 15:30 - 16:30 At the dentist: Fixing my teeth with caries? (Yes / Y)

<< Yes

Event was cancelled

Please choose what to do (1 - Create an event | 2 - Daily schedule | 3 - Search event | 4 - Find availability)

<<4

Enter date:

<< 20.06.2020

From time:

<< 13:00

Enter duration:

<< 2

Available slot for the event: 20.06.2020 14:30

6. Система за отпуски

Създайте система, в която служителите на компания могат да заявяват отпуски. Системата да се реализира в конзола. При стартиране на програмата излиза следното меню:

1. Заяви отпуски
2. Виж всички отпуски

3. Виж отпуска за служител
 4. Промени статус на отпуска
 5. Изход
-

Въведи избор:

- При избиране на 1, потребителят въвежда последователно Име, имейл, ЕГН, две дати - за начало и край на отпуската, тип на отпуската - платена или неплатена. При грешен формат на данните да излизат подходящи съобщения.
- При избиране на 2, на екрана се показват във формата на таблица, всички направени до сега заявки. Всички заявки се помнят при изключване на програмата.
- При избиране на 3, потребителят въвежда името на служител и на екрана излиза таблица, но само със заявките за този служител.
- Всички заявки да си имат уникален номер. Към таблиците от опция 2 и 3 да се добави и нова колона "Статус". Статусът може да бъде approved, rejected or pending. При избиране на 4, излиза таблица подобна на опция 2, но с допълнителна колона - номер на заявката. Потребителят може да въведе номер на заявка и статус, което ще промени статуса на съответната заявка.

7. Бесеница

Бесеница е името на популярната в цял свят игра за познаване на думи. Целта ви е да отгатнете скритата дума, чиито букви са отбелязани с чертички. За целта в конзолата се въвежда буква от азбуката. Ако буквата я има в намислената дума, тя ще се покаже на съответното място (или места, ако се среща повече от веднъж). В противен случай на бесилката се рисува част от тялото на човечето. Човечето се изобразява в конзолата. Когато то се нарисува изцяло, играта приключва - човечето е "обесено"! Нека броят на грешките за всяка дума да бъде един и същ.

Играта да работи с речник, съставен от имената на населените места в България.

Играта трябва да има и вариант да се играе от двама души. Те записват своите имена в началото. Първият играч познава буква, ако познае, продължава да познава втора буква. Редът му свършва като направи грешка. След това е вторият играч, който също познава букви до първа грешка. Който реши думата напълно е победител и печели 1 точка. Играта пита дали играчите искат да продължат да играят. Ако да, играта продължава с нова дума, ако не, играта изкарва броя точки на играчите.





ГОЛЯМА ЖЕЛЯЗНА

8. Викторина

Викторината е забавна състезателна игра на отговори на въпроси, обикновено обединени в обща тема. Играта се играе текстово в конзолата. Целта ѝ е да отговорите на възможно най-много последователни въпроси правилно. При грешен отговор играта се прекратява и се изписва текущият резултат на конзолата.

За целите на проекта ще работим със csv структура запазена в стринг, от който ще се четат въпросите и отговорите. За всяка категория трябва да има кореспондиращ стринг, където да се пазят въпросите и отговорите. Форматът на всеки стринг ще бъде следния:

Въпрос1, ПравilenОтговор, ГрешенОтговор1, ГрешенОтговор2,
ГрешенОтговор3

Въпрос2, ПравilenОтговор, ГрешенОтговор1, ГрешенОтговор2,
ГрешенОтговор3

Важно е да се отбележи, че всеки въпрос е на отделен ред в стринга и за разделител се използва символът запетайка.

В началото на играта потребителят може да избира от няколко категории, в които може да играе. След избор на категория, започват да се появяват въпросите. Те от своя страна всеки път трябва да се визуализират в произволен ред. Отговорите всеки път също трябва да се показват по произволен начин. Ето пример как трябва да се показват в конзолата:

Въпрос:

1. Отговор1
2. Отговор2
3. Отговор3
4. Отговор4

При правилно избран отговор, се появява случайно избрания следващ от съответната категория. При грешен - категорията, в която сме играли, и резултатът до момента. Играта се прекратява.

9. Крави и бикове

Игра, в която участникът трябва да познаеш намислено число. Играе се по следния начин:

- Компютъра намисля число
- Играча въвежда число
- Компютъра извежда колко крави и бика има в числото
- Играча въвежда ново число
- Компютъра извежда колко крави и бика има в новото число
- Играе се докато играча не познае числото

Играчът има крава, ако във въведеното от него число има цифра, която я има в намисленото

Играчът има бик, ако във въведеното от него число има цифра, която я има в намисленото и тази цифра е на същата позиция в двете числа

Ограничения: намисленото число трябва да е четирицифрено, с различни цифри.

Играта трябва да може да се играе и от двама души. При този вариант, компютъра си намисля две числа и играчите се редуват да познаят своето число. Играта приключва когато някой от двамата успее.

Пример:

- Намислено число: 8456
- Въведено предположение: 4231
- На екрана се изписва: 1 крава
- Въведено предположение: 4512
- На екрана се изписва: 2 крави
- Въведено предположение: 5421
- На екрана се изписва: 1 крава, 1 бик
- и т.н.

10. Матрици

- въвеждане на матрица от клавиатурата
 - да се задават размерности и да се въвеждат елементите по удобен начин
- извеждане на матрица на екрана
 - да се извежда матрицата в подреден табличен вид
- събиране и изваждане на матрици (на 2 матрици)
- умножение на матрици

- да се прави проверка дали размерностите на матриците са подходящи
- намиране на детерминанта на матрица
 - без значение кой алгоритъм ще бъде използван
- намиране на обратна матрица
 - да се извежда грешка, ако матрицата няма обратна
- проверка дали дадена квадратна матрица може да се преобразува до единичната (E)

11. Хотел

Да се напише компютърна програма, реализираща информационна система, обслужваща хотел. Програмата да поддържа текстов диалогов режим, позволяващ удобен интерактивен избор на следните операции:

- регистриране в стая (задава се номер на стая, начална и крайна дата и се въвежда коментар, например "семејство Симпсън")
- списък на свободни стаи
- освобождаване на заета стая
- справка за използването на стаи в даден период (по начална и крайна дата се извежда списък, в който за всяка стая, използвана в дадения период, се извежда и броя на дните, в които е била използвана)
- намиране на подходяща (с необходимия брой легла) свободна стая по дадена начална и крайна дата
- задаване на допълнителни изисквания за стаи (бебешка кошарка, със/без закуска, с изглед към морето/планината/двора)

Пример:

Please select what you want to do (1 - Make a reservation; 2- List free rooms; 3 - Checkout room; 4 - Stats; 5 - Find a room; 6 - Update a room)

<< 1

Enter room number:

<< 105

Start date:

<< 12.04.2021

End date:

<< 16.04.2021

Enter notes:

<< Petrovi

Reservation was created!

Please select what you want to do (1 - Make a reservation; 2- List free rooms; 3 - Checkout room; 4 - Stats; 5 - Find a room; 6 - Update a room)

<< 4

Start date:

<< 10.03.2021

End date:

<< 15.0.3.2021

- 101: 5 days
- 102: 5 days
- 103: 0 days
- 104: 2 days
- 105: 1 day

Please select what you want to do (1 - Make a reservation; 2- List free rooms; 3 - Checkout room; 4 - Stats; 5 - Find a room; 6 - Update a room)

<< 5

Enter number of beds:

<< 3

Enter start date:

<< 20.02.2021

Enter end date:

<< 23.02.2021

Available rooms:

- 102
- 103

12. Монополи*

Монополи е игра, състояща се в добиването на печалби чрез икономическа дейност, като купуване, наемане и отдаване на имоти с помощта на игрални "пари", докато се хвърлят зарове и се придвижват по игралното поле. Играта се играе текстово в конзолата, без да се чертае дъска. Секторите на игралното поле "Монополи" са общо 40 и всяко от тях изисква играчите да извършат определено действие. Секторът "Начало" е мястото, от което играчите започват участието си в "Монополи". При стартиране на играта:

- избира се броят на играчите (максимум 4)
- въвеждат се имена за играчите
- играта подканя играч да хвърля двата зара (пример: Лили ти си на ход! Натисни „enter“, за да хвърлиш заровете)

- след хвърлянето играта ти казва колко си хвърлил и къде се намираш след преместването
- пита те искаш ли да купиш/наемеш/продадеш имот или не
- ако полето е специално може да се изтегли карта или попадаш в затвор. Ако си в поле с чужд имот дължиш пари на собственика или фалираш и губиш играта.

Стъпките се повтарят за всеки играч. Всеки път, когато играча премине през старта получава заплата.

<https://www.game.store.bg/g/id-43/monopoli-pravila-i-istoria-na-igrata.html>

* При изцяло работеща логика, разбиране на кода и спазени Clean code правила курсиста получава 5 бонус точки към общия резултат.