### Main topics

შეიძლება რაღაცები გამომრჩა, მთელი განვლილი მასალა შედის, სლაიდებს გადაუყევით მაინც.

1. Class - დამოუკიდებლად უნდა შეგეძლოთ აღწერილი კლასის დაწერა და შემდეგ ამ კლასის ობიექტის შექმნა და გამოყენება. კარგი იქნება, რომ ქვემოთ ჩამოთვლილი სამივე ვარიანტი დაწეროთ
   1. data structure class (თქვენით დაწეროთ არსებული მონაცემთა სტრუქტურის ანალოგი მგ: Map, Set, Queue, LinkedList)
   2. custom class(კლასი რომელის რაღაც კონკრეტული ამოცანის გადასაჭრელადაა საჭირო. მგ: problem71- [School.java](https://github.com/vashli/MethodologiesSeminarSolutions/blob/main/src/School.java))
   3. implements Interface(უნდა იცოდეთ რა არის ინტერფეისი, როგორ შექმნათ ინტერფეისი და დაწეროთ კლასი, რომელიც არსებულ ინტერფეისს აიმპლემენტირებს მგ: Practice Problem - 2)
2. Data Structures
   1. array
   2. matrix
   3. arrayList
   4. map
   5. set
   6. iterator (get and use iterator with known data structures)
3. String manipulation - easy და medium ამოცანები შეგიძლიათ გააკეთოთ [leetcode](https://leetcode.com/problemset/all/?topicSlugs=string&page=1)-ზე
4. Interactors - წიგნში კარგად წერია და იქ ჯობია გადახედოთ, თავი - 10.6 The Swing interactor hierarchy + 10.7 , კოდის მაგალითები [InteractorSample](https://github.com/vashli/MethodologiesSeminarSolutions/blob/main/src/InteractorsSample.java), [Problem70](https://github.com/vashli/MethodologiesSeminarSolutions/blob/main/src/Problem70.java)
5. Graphics
   1. animations ([MidtermPractice](https://docs.google.com/document/d/1AVkTzyPBdkFZYUKBjRnxR0tspiycwlnID53uKB-6XKE/edit#)-ის ანიმაციის ან წინა შუალედურების მაგალითები გააკეთეთ)
   2. interactors
   3. list of graphical objects ( ბევრი ერთი ტიპის ობიექტის შექმნა, განლაგება, და ერთიანად გამოძრავება უნდა შეგეძლოთ , მგ: Practice Problem - 6))
6. RandomSimulations ([MidtermPractice](https://docs.google.com/document/d/1AVkTzyPBdkFZYUKBjRnxR0tspiycwlnID53uKB-6XKE/edit#)-დან ერთი ამოცანა გაარჩიეთ ამ თემაზე)

### 

### 

### Main Topics Review Playlist

კლასები და გრაფიკები არაა და აუცილებლად უნდა გაიმეოროთ ეგენიც

<https://www.youtube.com/watch?v=zjpmqXXrq-Y&list=PL6qg6DlVkjd5qcN9fP6DbzqmIugbiepWY&index=2>

### Other Problems

problem set:

<https://drive.google.com/file/d/18w3cRI78rzG9t4Ze9wSORbIoNdrbHg4f/view>

MidtermPractice:

<https://docs.google.com/document/d/1AVkTzyPBdkFZYUKBjRnxR0tspiycwlnID53uKB-6XKE/edit#>

BonusProblems:

<https://docs.google.com/document/d/1g11BWjcUWXw5E16peFSMZz2_Y7lOls7EeFJlzBoa2bA/edit#>

### 

### Practice Problems

### შუალედურის შემდეგ გავლილი თემები(სემესტრის I ნახევრის თემებისთვის ნახეთ [MidtermPractice](https://docs.google.com/document/u/0/d/1AVkTzyPBdkFZYUKBjRnxR0tspiycwlnID53uKB-6XKE/edit))

solutions at: <https://github.com/vashli/MethodologiesSeminarSolutions/tree/2022/src>

ფრჩხილებში წერია ამოხსნის კოდის ფაილის სახელი.

1. დაწერეთ კლასი myMap, რომელიც აიმპლემენტირებს Map interface-ს. კლასის შიგნით მონაცემების შესანახად შეგიძლიათ გამოიყენოთ ნებისმიერი ტიპის მონაცემთა სტრუქტურა თვითონ map-ის გარდა. ატვირთულ კოდში generic კლასია მოცემული. თქვენ შეგიძლიათ String-ების მეპი დაწეროთ სიმარტივისთვის(MyMap, MyMapTst)

Tags: Class, Interface, DataStructure, Map

1. დაწერეთ კლასი MidtermBonus , რომელიც აიმპლემენტირებს [BonusInterface](https://github.com/vashli/MethodologiesSeminarSolutions/blob/main/src/BonusInterface.java)-ს. მოსწავლე აკეთებს guess-ს და ირჩევს მთელი რიცხვს 0-

დან 100-მდე შუალედში(0-ის და 100-ის ჩათვლით). winningNumber არის ყველა მოსწავლის მიერ არჩეული რიცხვების საშუალო

არითმეტიკული გავამრავლებული 2/3-ზე და აღებული მთელი ნაწილი.

winners არიან მოსწავლეები, რომელთა guessიც იყო winningNumber-ის ტოლი.

ასევე დაწერეც ConsoleProgram კლასი რომელშიც MidtermBonus ობიექტს შექმნით და გატესტავთ.(MidtermBonus, BonusInterface, BonusConsole)

Tags: Class, Interface

1. problem set - 71 (School)

Tags: Class, Map

1. problem set - 72 ჯობია, რომ ჯერ 71 დაწეროთ და მერე დაიწყოთ ამისი წერა. მაგრამ მარტო ამაზე მუშაობა თუ გინდათ, გითჰაბიდან წინას ფაილი ჩააკოპირეთ იგივე ფოლდერში რომ School კლასის გამოყენება შეძლოთ (Problem72)

Tags: Interactors, Graphics, Custom Objects

1. დაწერეთ გრაფიკული პროგრამა, რომელიც იქნება Map<String, String> - ის ვიზუალიზაცია. canvas-ის სამხრეთით დაამატეთ ორი textField: key, value

სამი ღილაკი: add, remove, get.

კანვასის მარცხენა მხარეს მუდმივად უნდა ჩანდეს map-ის ახლანდელი მდგომარეობა - ყოველი key-value წყვილი ახალ ხაზზე.

get ღილაკზე დაჭერის შემდეგ კანვასის მარჯვენა მხარეს უნდა გამოჩნდეს key value წყვილი, რომელიც შეესაბამება textField key- ში შეყვანილ ტექსტს

add ღილაკზე დაჭერისას ემატება textField-ში key-value წყვილი

remove ღილაკზე დაჭერისას თუ value Field ცარიელია, უნდა წაიშალოს key field-ის შესაბამისი ელემენტი, თუ სავსეა უნდა წაიშალოს ისეთი ელემენტი , რომელიც key - value წყვილს შეესაბამება

ღილაკზე დაჭერის შემდეგ field-ები გაასუფთავეთ.

Tags: Interactors, Graphics, Map

1. დაწერეთ გრაფიკული პროგრამა რომელიც იქნება სქროლვადი სიის იმპლემენტაცია. canvas-ის სამხრეთით დაამატეთ textField: value

ორი ღილაკი: add, remove.

canvas-ზე მუდმივად უნდა ჩანდეს სიის ახლანდელი ვერსია.

add ღილაკზე დაჭერისას ემატება value field-ში არსებული სტრინგი სიას.

remove ღილაკზე დაჭერისას value field-ში არსებული სტრინგის პირველი შეხვედრა იშლება სიიდან.

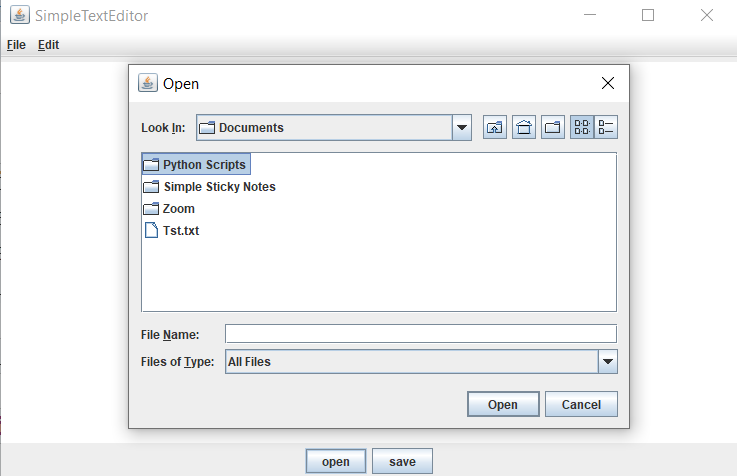
კანვასის მარკვენა კიდეში უნდა დაამატოთ მართკუთხედი რომელიც შეგიძლიათ გაამოძრაოდ მაუსის დრაგით. შესაბამისად უნდა მოძრაობდეს გრძელი სიაც ისე, რომ ყოველთვის შეგვეძლოს სიის ყველა ელემენტის დათვარიელება. მოკლე სიისას თუ გინდათ არ გამოაჩინოთ ობიექტი, ან სია არ მოძრაობდეს (Final06)



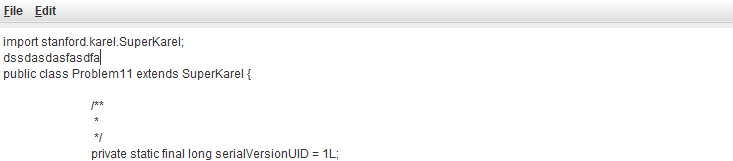
Tags: Interactors, Graphics, Scroll, ArrayList, Animations, List of GObjects

1. დაწერეთ SimpleTextEditor(© r.meishvili): პროგრამა უნდა შეიცავდეს მთავარ ტექსტის არეას, სამხრეთით ორ ღილაკს: open, close.

open-ზე დაჭერისას უნდა გამოვარდეს შემდეგი ფანჯარა: გამოიყენეთ JFileChooser



მთავარ ფანჯარაში უნდა შეგეძლოთ ტექსტის დაედიტება - JTextArea



Save - ზე დაჭერისას უნდა ინახავდეთ შეცვლილ ფაილს(Final07)

Tags: Interactors, File read/write

1. შექმენით HashMap<String, Integer> ობიექტი, შეავსეთ. ჩაწერეთ ფაილში. შემდეგ წაიკითხეთ ეგ ფაილი და წაკითხული მონაცემები შეინახეთ ახალ HashMap-ში. ფაილში როგორი სტრუქტურითაც გინდათ ისე შეინახეთ, მთავარია წაკითხვის მერე ანალოგიური map-ის შევსება მოახერხოთ სწორად. (Final08)

Tags: File read/write, Map

1. დაწერეთ მეთოდი რომელსაც არგუმენტად გადმოეცემა int-ების მატრიცა და ამ მატრიცას სორტავს.(Final09)

Tags: Matrix, Sort

1. დაწერეთ მეთოდი რომელსაც არგუმენტად გადმოეცემა int-ების მატრიცა და ამ მატრიცას ატრიალებს სიმეტრიულად y ღერძის მიმართ (Final10)

1, 2 -> 2,1

3, 4 4,3

Tags: Matrix, Rearrange elements

1. დაწერეთ მეთოდი რომელსაც არგუმენტად გადმოეცემა int-ების მატრიცა და ამ მატრიცას ატრიალებს სიმეტრიულად x ღერძის მიმართ (Final11)

1, 2 -> 3,4

3, 4 1,2

Tags: Matrix, Rearrange elements

1. დაწერეთ მეთოდი რომელსაც არგუმენტად გადმოეცემა int-ების მატრიცა და ამ მატრიცას უკეთებს ტრანსპონირებას (Final12)

Transpose of a matrix is a matrix which is formed by turning all the rows of the source matrix into columns and vice-versa.

[[1, 2, 3, 4], [[1, 5, 9, 3],

[5, 6, 7, 8], [2, 6, 0, 4],

[9, 0, 1, 2], —------> [3, 7, 1, 5],

[3, 4, 5, 0]] [4, 8, 2, 0]]

Tags: Matrix, Rearrange elements

1. დაწერეთ მეთოდი - private void randomSort(ArrayList<Integer> list) რომელსაც გადაეცემა integer-ების arrayList და რომელიც ამ სიის ელემენტებს გადაალაგებს რანდომ თანმიმდევრობით. ანუ შემთხვევითი მიმდევრობის ამოსვლის ალბათობა ყველა სხვა მიმდევრობასთან შედარებით თანაბარ ალბათური უნდა იყოს.(Final13)

Tags: ArrayList, Custom Sort, Random

1. დაწერეთ მეთოდი, რომელსაც გადმოეცემა სტრინგების ლისთი და სორტავს ამ ლისთს სტრინგების სიგრძის მიხედვით კლებადობით(Final14)

Tags: Algorithm, Sort, Custom order

1. გაქვთ HashMap<String, String> studentTeachers, სადაც ვინახავთ მოსწავლე -> მასწავლებლის წყვილებს. ყველა მოსწავლეს ერთი მასწავლებელი ჰყავს, ბევრ მოსწავლეს შეიძლება იგივე მასწავლებელი ჰყავდეს.

დაწერეთ boolean მეთოდი validateTeachers, რომელსაც გადაეცემა ეს მეპი და ამოწმებს რომ არც ერთი მოსწავლის წინაპარი მასწავლებელი(მასწავლებლის მასწავლების… მასწავლებელი) არ გამოდის საკუთარი თავი.(Final15)

Tags: Algorithm, Maps

1. დაწერეთ მეთოდი Merge, რომელსაც გადმოეცემა ორი დასორტილი ArrayList და რომელიც აბრუნებს ახალ ArrayList -ს , რომელიც არის არგუმენტად გადმოცემული სიების გაერთიანება დასორტილ მდგომარეობაში.

მეთოდი დაწერეთ შემდეგი ალგორითმით:

შემოიტანეთ ცვლადი პირველი მასივის ახლანდელი ინდექსის აღსანიშნავად, და მეორე ცვლადი მეორე მასივის ახლანდელი ინდექსის აღსანიშნავად.

ყოველ ჯერზე შეადარეთ ახლანდელ ინდექსებზე არსებული ელემენტები და მათ შორის მინიმალური დაამატეთ საბოლოო სიაში. გადატანილი ცვლადის შესაბამისი ინდექსის აღმნიშვნელი ცვლადი გაზარდეთ და განაგრძეთ პროცესი. თუ რომელიმე სიის ბოლოში გახვედით, მეორე სიის დარჩენილი ნაწილი ავტომატურად უნდა დაამატოთ საბოლოო სიას.(Final16)

Tags: Algorithm Implementation, Merge