

Main Assignment - CS12320

Individual Assignment: Question bank

Nikola Nikolov

nin6@aber.ac.uk

May 5, 2021

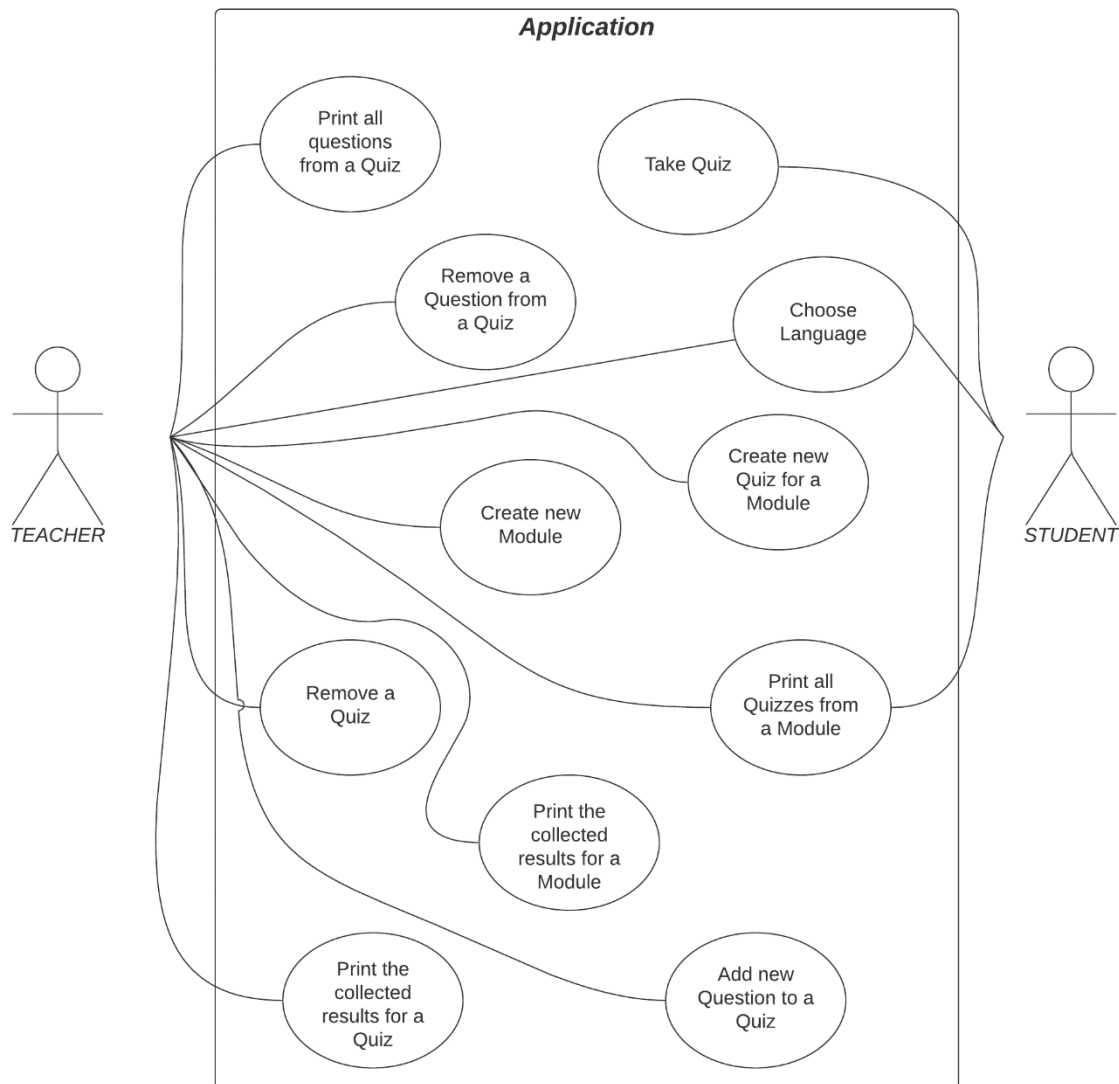
Contents

Contents.....	2
Introduction	3
Design.....	4
Questions	6
Bank	6
Application	6
Testing.....	7
Evaluation	12

Introduction

That is a report on what I have done to complete my assignment. Brief explaining of the functionality of the prototype program I had to make is, a program that allows to teachers to create quizzes of every module that they want, while the students can do the quizzes and see their results. In this report I will explain how I solved the problems that I faced making the program. Also, the way I write the code and why I did it as I did. There is going to be a part that will focus on the testing of the program and the things I learned while I was doing the program.

To help you with the understanding of my code I will put an UML use case diagram here in the beginning. On the UML use case diagram, you can see what the different kind of users can do and what they cannot.



After I finished the functional and non-functional requirement I decide to add a bit more flair just to be sure that the customer will be satisfied (the lecture will give me more points). And by flair I meant that I make the Application in way that can store the results of every student that have done the quiz and the teacher will have the chance to see a sample statistic of the results only for a specific quiz or for the whole module. Also, I have added one more type of question, match-type of question (there are two columns, and you have to connect every line with the correct one on the other column).

Design

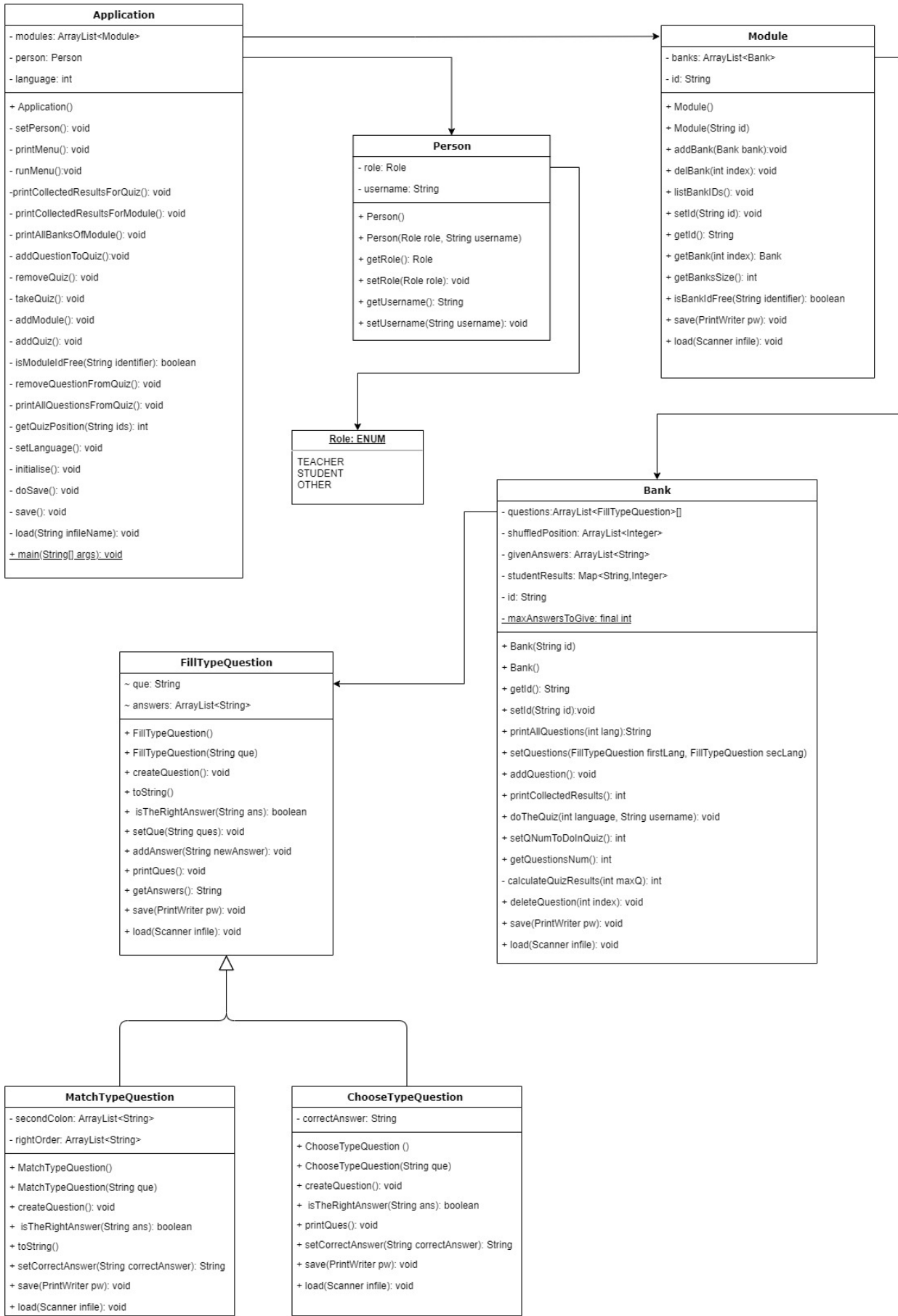
In this section we will look at the whey I created and connected the classes. That will be followed by explanation on every class individually and the functions that contains.

You can see the class diagram on the following page. However, I will put a SVG copy in the zip file just in case it is not readable enough because of the zooming.

At the beginning I was not sure what will be the best way to use inheritance for my question classes. I thought to create a super class called Question and all types of question to be inherited by it. But I realized that no matter how I do it I will need to override almost every method like createQuestion, isTheRightAnswer, printQues, even the save method, because all of them had to treat the received information differently and to print different thing. So I decide not to create a Question class which is going to have just to variables in it and unusable methods, Instead I made the Fill-TypeQuestion the super class because that class have the two variables that every type of question will need. I thought that creating that class will be unnecessary and I make me type more code for nothing, but still, I am not sure in that decision. Please let me know if the other way was better and why if so.

All type of questions are inherited by my Fill-TypeQuestion class which is connected to the Bank class. The Bank class can every type of questions stored in the array called "questions". "questions" is an array of ArrayList, it is that why so it can store as much language as needed. For now, it is set to two as the requirement asked, but it can easily de set to what every number just with little corrections in the Bank class.

The Module class have an ArrayList of Bank, in same way as Application class have ArrayList of Modules. The Application class is the class that make the corresponding with the use in way so the use can choose which method to activate. The Application class have variable of type Person too. The person class is small but is maid so the program to have identifier like username and the role of the person. When the security is ad that class can expand and more useful. I decide that the Enum class to choose the role is a good Idea just because in future updates is likely to have more roles like directors, admins, or others.



Questions

The question classes are not that different from each other. They have pretty much the same methods, but the methods work in different way. For example, print method for Fill-type questions must print only the sentence while Choose-type of question must print the answers as well.

In this classes an interesting method is the is theRightAnswer. That method receives information in a string, but every class treats the information in different way. For example, if the question is of choose to type the method will take the first char (it must be just one) and will cast the letter into a number in way that A will be equal to 0, B to 1 and so on. After that, it will check if on the place is the correct answer. But if the question is of fill type the same function will just loop through the answer and if there is a match it will return true.

Bank

The most complex method in this class is doTheQuiz. After initializing everything that will be needed in the class the setQNumToDoInQuiz method which ask the use how many questions will do from that quiz. After that, every time we check if the responds of the user are the same as the special characters that will move to next or previous question or submit the work. If the respond is not match for none of these it is going to be save as an answer for the question. When the user submits its work the calculateQuizResults is called. That method counts how many of the question ware left without and answer and print the num at the end. Meanwhile, it counts how many of the answers are correct by calling theRightAnswer method from the respective class. At the end prints the result in percentage and save the result and the username of the use.

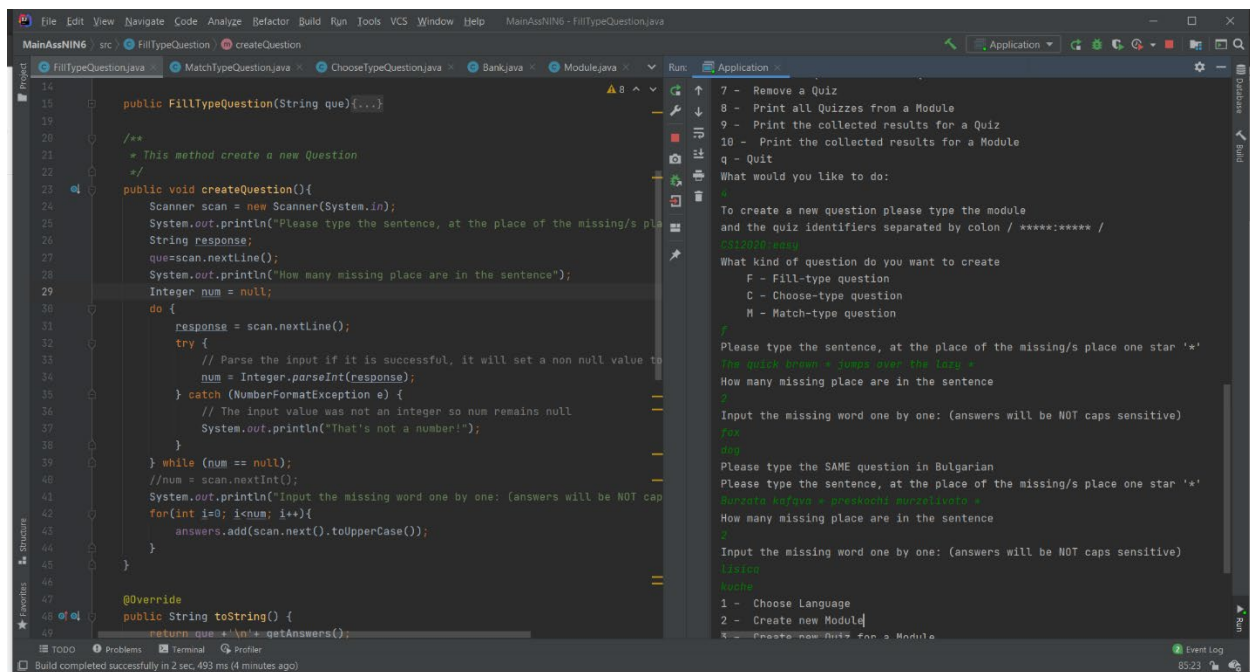
Application

The other worth explaining class is Application. The RunMenu method allows the user to use the methods that are purpose for them. Every use is separate in different method in way that they can be easy to understand.

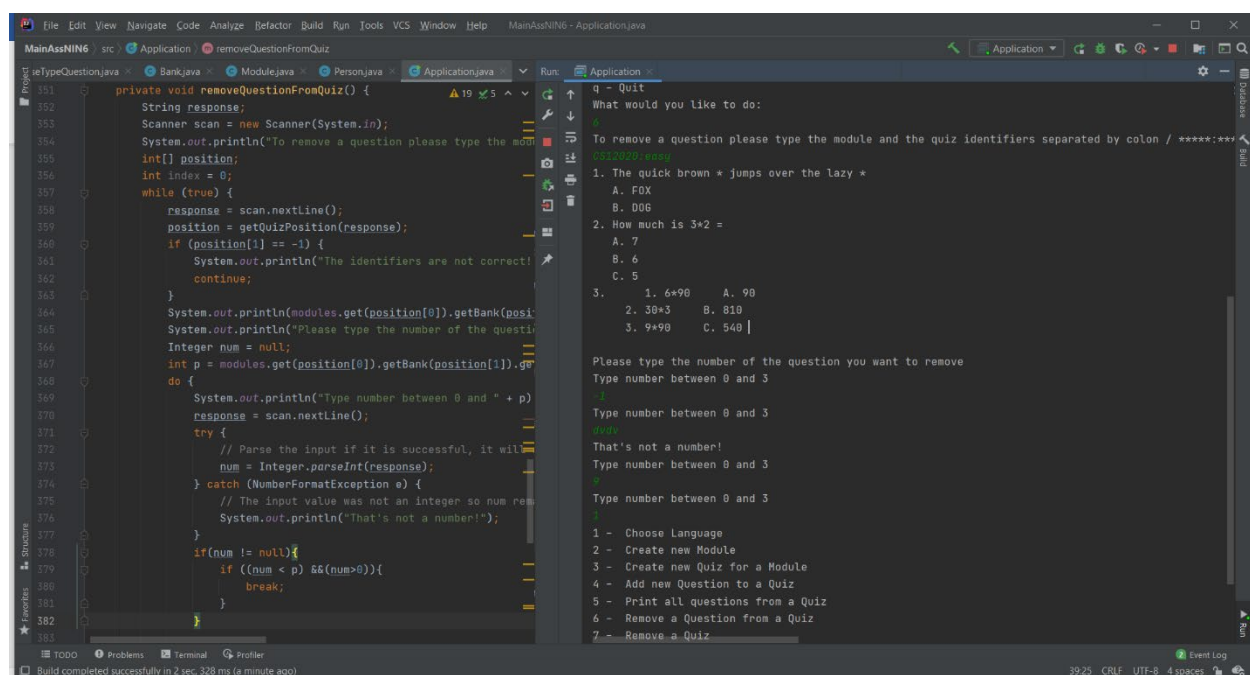
Testing

ID	Requirement	Description	Inputs	Expected outputs	Pass/ Fail	Comments
A1.1	FR1	Every module have unique ID	Module's 2 ID to be same as the id of the first one	Return error message and ask for new id because the id is already used	P	
A1.2	FR2	Add a question to a bank	Input the que = "How toll am I?" And answers to choose from	The question and the answers to be saved properly	P	
A1.3	FR3	Removing Question	IDs of the Module and The Bank, and the number of the question	To remove just the question on that number in this Bank	P	
			Wrong Module ID	To ask for another ID	P	
			Number that is not in the range	To ask for new number	P	
A1.4	FR4	List all the question banks	The ID of the Module	To print the questions	P	
			Wrong Module ID	To ask for another ID	P	
A1.5	FR5	Delete a question bank	Empty Bank	To be deleted	P	
			Full Bank	NOT to be deleted and message to say that the bank has to be empty	P	
A1.6	FR6	Take a quiz	Correct response	No errors	P	
			Incorrect response	Asking for correct input	P	
A1.7	FR7	End the quiz at any time	Try to end it	End it	P	

A1.8	FR8	The quiz displays Q questions in a random order.	Enter number in range	Print the number of questions	P	
			Enter bigger number that the questions	Print the max of questions	P	
			Enter negative number	Ask for new number	P	
A1.9	FR9	Student decides when they want to move to other question	Try to move to the next question	To print the next question	P	
			Try to move to the previous question	To print the previous question	P	
			Try to go over the limits (use next q on the last one)	Print a message that that cannot be done	P	
A1.10	FR10	question is displayed in an appropriate textual	--	question is displayed in an appropriate textual	P	
A2.1	NFR1	question bank must be persistent and stored in a text-based database	-	-	P	
A2.2	NFR2	text-based, menu-driven user interface	-	text-based, menu-driven user interface	P	
A2.3	NFR3	Two languages are supported	Choose language	Print the questions is chosen language	P	
A2.4	NFR4	possible to plug in new kinds of question	-	-	P	But little correction on the bank class will be necessary



A1.2 I have input question.



A1.3 Removing question - Trying to put incorrect numbers.

```
172
173
174 private void printCollectedResultsForModule() {...}
175
176
177
178 /** This method prints the id of every bank in a module */
179 private void printAllBanksOfModule() {
180     System.out.println("Type id of the Module:");
181     String response;
182     Scanner scan = new Scanner(System.in);
183     do {
184         response = scan.nextLine();
185         if(isModuleIdFree(response)){
186             System.out.println("That id do not match.Try again!");
187         }else{
188             break;
189         }
190     } while (true);
191     for (int i = 0; i < modules.size(); i++) {
192         if (modules.get(i).getId().equals(response))
193             modules.get(i).listBankIDs();
194     }
195 }
196
197 /** This method add a question to a bank(Quiz) */
198 private void addQuestionToQuiz() {
199     String response;
200     Scanner scan = new Scanner(System.in);
201     System.out.println("To create a new question please type the module id and the quiz identifier separated by colon");
202     int[] position;
203     while (true) {
204         response = scan.nextLine();
205     }
206 }
```

1 - Choose Language
2 - Create new Module
3 - Create new Quiz for a Module
4 - Add new Question to a Quiz
5 - Print all questions from a Quiz
6 - Remove a Question from a Quiz
7 - Remove a Quiz
8 - Print all Quizzes from a Module
9 - Print the collected results for a Quiz
10 - Print the collected results for a Module
q - Quit
What would you like to do:
3
Type id of the Module:
0512020
That id do not match.Try again!
0512020
easy
hara
1 - Choose Language
2 - Create new Module
3 - Create new Quiz for a Module
4 - Add new Question to a Quiz
5 - Print all questions from a Quiz
6 - Remove a Question from a Quiz
7 - Remove a Quiz
8 - Print all Quizzes from a Module
9 - Print the collected results for a Quiz
10 - Print the collected results for a Module
q - Quit
What would you like to do:

A1.4 List all banks - Trying to put incorrect id.

```
235
236
237
238 /** This method remove a bank(Quiz) from a module */
239 private void removeQuiz() {
240     String response;
241     Scanner scan = new Scanner(System.in);
242     System.out.println("To delete Quiz please type the module id and the quiz identifier separated by colon");
243     int[] position;
244     while (true) {
245         response = scan.nextLine();
246         position = getQuizPosition(response);
247         if (position[0] == -1 || position[1] == -1) {
248             System.out.println("The identifiers are not correct");
249             continue;
250         }
251         if (modules.get(position[0]).getBank(position[1]).getQuiz().contains(response)) {
252             modules.get(position[0]).delBank(position[1]);
253         } else {
254             System.out.println("The Quiz has to be have NO questions inside it");
255         }
256         break;
257     }
258 }
259
260 /** This method starts the quiz(bank) */
261 private void takeQuiz() {...}
262
263 /** This method add new module */
264 private void addModule() {...}
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

What would you like to do:
3
To witch module do you want to add quiz. Please type the identifier:
0512020
What the quiz identifier will be:
hara
1 - Choose Language
2 - Create new Module
3 - Create new Quiz for a Module
4 - Add new Question to a Quiz
5 - Print all questions from a Quiz
6 - Remove a Question from a Quiz
7 - Remove a Quiz
8 - Print all Quizzes from a Module
9 - Print the collected results for a Quiz
10 - Print the collected results for a Module
q - Quit
What would you like to do:
7
To delete Quiz please type the module and the quiz identifiers separated by colon / ***** /
The Quiz has to be have NO questions inside it
0512020:hara
1 - Choose Language
2 - Create new Module
3 - Create new Quiz for a Module
4 - Add new Question to a Quiz
5 - Print all questions from a Quiz
6 - Remove a Question from a Quiz
7 - Remove a Quiz
8 - Print all Quizzes from a Module
9 - Print the collected results for a Quiz
10 - Print the collected results for a Module
q - Quit
What would you like to do:
7

A1.5 Remove an empty bank

```
138 /** By this method the user can do a quiz ...*/
143 public void doTheQuiz(int language, String username) {
144     LocalDateTime now = LocalDateTime.now();
145     givenAnswers = new ArrayList<String>();
146     for (int i = 0; i < maxAnswersToGive; i++) givenAnswers.add(
147         int currentQues = 1;
148         String response;
149         Scanner scan = new Scanner(System.in);
150         Collections.shuffle(shuffledPosition);
151         int maxQtoAnswer = setNumToDoInQuiz();
152         printQuizMenu();
153         long start = System.currentTimeMillis();
154         do {
155             System.out.println(currentQues + "/" + maxQtoAnswer);
156             System.out.print(currentQues + " ");
157             questions[language].get(shuffledPosition.get(currentQues));
158             response = scan.nextLine().toUpperCase();
159             switch (response) {
160                 case "N":
161                     if (currentQues == maxQtoAnswer) {
162                         System.out.println("That is the last question");
163                         break;
164                     }
165                     currentQues++;
166                     break;
167                 case "P":
168                     if (currentQues-1 != 0) {
169                         System.out.println("That is the first question");
170                         break;
171                     }
172                     currentQues--;
173                     break;
174                 case "S":
175                     // ... (rest of the code) ...
176             }
177         } while (!response.equals("S"));
178         long end = System.currentTimeMillis();
179         float time = (end - start) / 1000;
180         int hour = (int) (time/120); time = time-hour*120;
181         int min = (int) (time/60); time = time-min*60;
182         System.out.println("You do the quiz for " + hour + ":" + min);
183         int result = calculateQuizResults(maxQtoAnswer);
184         System.out.println("Your Result is " + result + "% !");
185         studentResults.put(username, result);
186     }
187 }
188
189 /** This method asks the user how many questions he wants to answer ...*/
190 public int setNumToDoInQuiz() {
191     // ... (rest of the code) ...
192 }
193
194 /** Get the num of questions ...*/
195 public int getQuestionsNum() { return questions[0].size(); }
196
197 /** This method returns the result of the quiz in % ...*/
198 private int calculateQuizResults(int maxQ) {
199     // ... (rest of the code) ...
200 }
201
202 /** This method remove a question ...*/
203 public void deleteQuestion(int index) {
204     // ... (rest of the code) ...
205 }
```

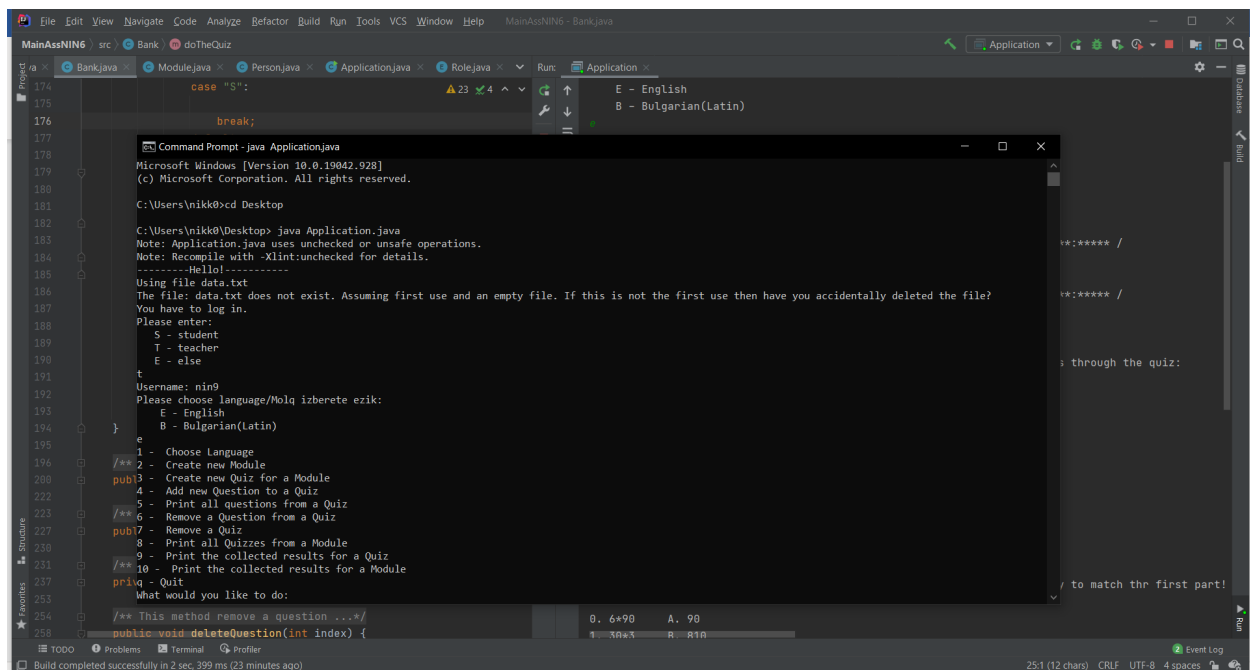
3 - Print all Quizzes from a Module
q - Quit
What would you Like to do:
Please type the module and the quiz identifiers separated by colon / ***** /
How many questions from the quiz you want to do
Before you start the quiz you have to know that you can use this commands through the quiz:
N - next question (automatically saves the answer)
P - previous question (automatically saves the answer)
S - Submit & Exit
1/2
1. Choose the correct answer. Type just the letter before the answer!
How much is 3*2 =
A - 7
B - 6
C - 5
0
2/2
2. Complete the sentence.
The quick brown * jumps over the lazy dog
for
That is the last question. You can submit your work by typing 'S'
2/2
2. Complete the sentence.
The quick brown * jumps over the lazy dog
You do the quiz for 0:0:19
0 not answered questions
Your Result is 100% !
Choose language

A1.6/ A1.7/ A1.8 Ending the Quiz and seen the result

```
174 case "S":
175     break;
176 default:
177     givenAnswers.set(shuffledPosition.get(currentQues));
178     if (currentQues == maxQtoAnswer) {
179         System.out.println("That is the last question");
180         break;
181     }
182     currentQues++;
183 }
184 } while (!response.equals("S"));
185 long end = System.currentTimeMillis();
186 float time = (end - start) / 1000;
187 int hour = (int) (time/120); time = time-hour*120;
188 int min = (int) (time/60); time = time-min*60;
189 System.out.println("You do the quiz for " + hour + ":" + min);
190 int result = calculateQuizResults(maxQtoAnswer);
191 System.out.println("Your Result is " + result + "% !");
192 studentResults.put(username, result);
193 }
194
195 /** This method asks the user how many questions he wants to answer ...*/
196 public int setNumToDoInQuiz() {
197     // ... (rest of the code) ...
198 }
199
200 /** Get the num of questions ...*/
201 public int getQuestionsNum() { return questions[0].size(); }
202
203 /** This method returns the result of the quiz in % ...*/
204 private int calculateQuizResults(int maxQ) {
205     // ... (rest of the code) ...
206 }
207
208 /** This method remove a question ...*/
209 public void deleteQuestion(int index) {
210     // ... (rest of the code) ...
211 }
```

Before you start the quiz you have to know that you can use this commands through the quiz:
N - next question (automatically saves the answer)
P - previous question (automatically saves the answer)
S - Submit & Exit
1/2
1. Choose the correct answer. Type just the letter before the answer!
How much is 3*2 =
A - 7
B - 6
C - 5
0
2/2
2. Connect the correct answer. Type just the letters without space in way to match the first part!
0. 6*90 A. 90
1. 30*3 B. 810
2. 9*90 C. 540
0
That is the last question. You can submit your work by typing 'S'
2/2
2. Connect the correct answer. Type just the letters without space in way to match the first part!
0. 6*90 A. 90
1. 30*3 B. 810
2. 9*90 C. 540
0
1/2
1. Choose the correct answer. Type just the letter before the answer!
How much is 3*2 =

A1.9 Moving to next and previous question

The screenshot shows an IDE window titled 'MainAssNIN6 - Bank.java'. The main editor displays Java code for a quiz application. A 'Command Prompt - java Application.java' window is open, showing the execution of the program. The command prompt output includes: 'Microsoft Windows [Version 10.0.19042.928] (c) Microsoft Corporation. All rights reserved. C:\Users\nikk0>cd Desktop C:\Users\nikk0\Desktop> java Application.java Note: Application.java uses unchecked or unsafe operations. Note: Recompile with -Xlint:unchecked for details. -----Hello!----- Using file data.txt The file: data.txt does not exist. Assuming first use and an empty file. If this is not the first use then have you accidentally deleted the file? You have to log in. Please enter: S - student T - teacher E - else t Username: nin9 Please choose language/Molq izberete ezik: E - English B - Bulgarian(Latin) e 1 - Choose Language 2 - Create new Module 3 - Create new Quiz for a Module 4 - Add new Question to a Quiz 5 - Print all questions from a Quiz 6 - Remove a Question from a Quiz 7 - Remove a Quiz 8 - Print all Quizzes from a Module 9 - Print the collected results for a Quiz 10 - Print the collected results for a Module priv 10 - Print the collected results for a Module What would you like to do: /** This method remove a question ...*/ public void deleteQuestion(int index) { 0. 6*90 A. 90 1 - A*1* 10 - A*1* Build completed successfully in 2 sec, 399 ms (23 minutes ago) 25:1 (12 chars) CRLF UTF-8 4 spaces Event Log

Starting the program from the command prompt.

Evaluation

I started by reading the assignment two or three times so I can memorize most of it. I left it for two days and try to imagine have everything will be connected and will how it will work. While I was working some ideas pop up and carefully collected them. After that period, I read the assignment two more times to be sure that I am not missing anything. I crated a class diagram in my textbook. That diagram was far from perfect but good enough to help me with the implementation of the program. I start from the question classes. The bank and the module followed shortly after that. At this point I have been made the classes with gutters and setters and some printing methods as well. Then I implement the Application class. First, I made the print menu class and the runMenu method. I put comments that explain what every swift case must do. Starting from the easiest one I cleated a private method in the Application class that called the right methods from the other classes. While I was expanding that class, I wrote some methods in the other classes in way that every function works properly.

There were some problems that was a bit difficult for me, but I googled them and read why and how other programs made a solution. That really helped me to expand my knowledge and to finish the assessment.

I decide for flair to add one more type of question and storing student results to monitor progress. I cannot assume have much my mark will be, because I tried me best and that is 90% on my knowledge. I am sure that there is better ways to do this program and the people that know this ways will mark my solution low. After all I hope that I receive 60-70%.