

	<pre> /** * Determina numarul de quiz-uri din lista pentru care s-a obtinut punctajul maxim. * Punctajul unui quiz este curprins intre 0 si 30. Lista de quiz-uri poate sa contina pana la 100 quiz-uri. * * @return numarul de quiz-uri pentru care s-a studentul a obtinut cel mai mare punctaj al sau. * Daca nu s-au efectuat quiz-uri atunci se returneaza 0. * Daca punctajul pentru fiecare quiz este 0 atunci se returneaza 0. * @throws QuizException - daca datele de intrare nu sunt valide */ </pre>
1	public int maxScoreQuizCounter() throws QuizException{
2	List<Quiz> quizList = allQuizzes();//preia din repository lista de quiz-uri
3	int index = 0, countMax = 0, posMax = 0;
4	if (quizList == null)
5	throw new QuizException(" null list ");
6	if (quizList.size()==0)
7	return 0;
8	if (!repository.getValidator().validate(quizList))
9	throw new QuizException(" data not valid ");
10	while (index<quizList.size()){
11	if (quizList.get(index).getCorrectAnswers() > quizList.get(posMax).getCorrectAnswers()){
12	posMax=index;
13	countMax=1;
	}
	else
14	if (quizList.get(posMax).getCorrectAnswers() == quizList.get(index).getCorrectAnswers())
15	countMax++;
16	index++;
	}
17	if (quizList.get(posMax).getCorrectAnswers() == 0)
18	return 0;
19	else return countMax;
20	}

	<pre> /** * Determina numarul de quiz-uri din lista pentru care s-a obtinut punctajul maxim. * Punctajul unui quiz este curprins intre 0 si 30. Lista de quiz-uri poate sa contina pana la 100 quiz-uri. * * @return numarul de quiz-uri pentru care s-a studentul a obtinut cel mai mare punctaj al sau. * Daca nu s-au efectuat quiz-uri atunci se returneaza 0. * Daca punctajul pentru fiecare quiz este 0 atunci se returneaza 0. * @throws QuizException - daca datele de intrare nu sunt valide */ </pre>
	public int maxScoreQuizCounter() throws QuizException{
	List<Quiz> quizList = allQuizzes();//preia din repository lista de quiz-uri
	int index = 0, countMax = 0, posMax = 0;
	if (quizList == null)
	throw new QuizException(" null list ");
	if (quizList.size()==0)
	return 0;
	if (!repository.getValidator().validate(quizList))
	throw new QuizException(" data not valid ");
	while (index<quizList.size()){
	if (quizList.get(index).getCorrectAnswers() > quizList.get(posMax).getCorrectAnswers()){
	posMax=index;
	countMax=1;
	}
	else
	if (quizList.get(posMax).getCorrectAnswers() == quizList.get(index).getCorrectAnswers())
	countMax++;
	index++;
	}
	if (quizList.get(posMax).getCorrectAnswers() == 0)
	return 0;
	else return countMax;
	}