web: suruse.uni-ruse.bg

BASIC USE CASES IN PROGRAMMING CONTESTS TRAINING SYSTEM

Galina Atanasova

Angel Kanchev University of Ruse

Abstract:*The article presents the place and the role of training systems in programming contests forums. Emphasis is placed on aproject of a programming contests training system. There are pointed basic user roles and their use cases in such a system. The described use cases may be used as basic set for programming contests system projects.*

Keywords: *Programming Contests, Programming Contests Training System,Use case.*

INTRODUCTION

The basis for success is always a lot of hard work and good preparation. The programming competitions are like all other competitions in terms of preparation needed to achieve results. A competitor can not be good without training. Solving many tasks is the only thing that can help in inventing and implementing algorithms and data structures. Also, experience helps overcome some types of often made mistakes. It remains to be solved the problem of the ability to train alone.

Some competitors may be lucky enough to have a teacher instructing them. Others may have a school to visit. This is a great start for those who want to become good in programming competitions, but it only helps somewhere. If someone wants to be among the best, he has to do much more than the time and the power can be taught by his lecturers. And there are things in which there is no way for a person to be only learned. It is important to make some conclusions yourself. So the effect is radically different.

So far, computer science is extremely popular and there are many places on the web where one can train alone. Besides the standard theory that can be learned by a teacher, a book or the Internet, they can be found much more useful - problems in which it can be applied [2].

Often, the problems in the training sites contain different features that make the problem solved more interesting or more complex. It's very useful to have seen some tricks before they hit an important competition.

The exercises practiced by the majority of contestants consist of two alternating stages: learning a new theory and solving problems related to it. The better a competitor becomes, the more the volume of new theory reduces, and the volume of solved tasks in each cycle must be increased accordingly. Competitive Informatics concentrates on certain popular themes and very rarely emerging tasks whose solution relies on another theory. By learning the specific topics, a competitor will be prepared for over 90% of the tasks that fall on a race.

EXISTING OPPORTUNITIES

There are systems that offer users the ability to practice and compete. Such are, for example https://www.hackerrank.com/ [3], http://mendo.mk/Welcome.do [4], http://www.ecs.csus.edu/pc2/, https://www.domjudge.org/ [1]._Typical of them is that the interface is in English and does not offer access to a set of systematized materials for preparation. Some of them, such as PC ^ 2, are complicated for configuring by a non-experienced user.

As a result of the study of existing alternatives and on the basis of the facts presented, the necessity of a system, which assists the training of computer science contestants, can be found. It is good to contain appropriate materials categorized by the theoretical type and resources for preparation and training for programming competitions. Potential users of this system will be students, students and lecturers to access structured training materials and training opportunities. Using the system will allow for adequate and easy preparation for participation in competitions and programming Olympiads.

FUNCTIONAL CHARACTERISTICS OF THE DESIGNED SYSTEM

The Programming contests software system should be implemented as a web application. It will provide its users with access to training materials, exemplary tasks accompanied by tests and author's decisions, and news related to racing programming. Training materials will be grouped into separate categories and marked with appropriate tags to allow competitors to navigate as easily and quickly as possible and to prepare them in a systematic manner. Sample tasks will be accompanied by tests and sample solutions so that the user can really see if a task is successfully resolved or is still working on it. The up-to-date news will contain information on upcoming events related to racing programming.

USER ROLES

The system's functional features are categorized into user roles that can perform or have access to these actions. Every role in the hierarchy inherits all the rights of the previous roles.

The following user roles have been identified:

Role	Access
Unregistered user	The unregistered user has very limited rights in the system.
Competitor	A student with an account in the system. She/he has right to view all objects in the system.
Lecturer	Teacher / lecturer with system account. She/he has the right to add objects and edit / delete the objects created by her/him in the system.
Administrator	System Administrator has the right to edit / delete all objects, as well as to perform all possible actions in the system.

INFORMATION OBJECTS AND OPPORTUNITIES

For each identified information object its characteristic features and actions are manipulated by a given user role.

Materials

Only lecturers have the opportunity to add training materials. The materials are divided into separate categories and marked with labels to make the information well organized. This way, users can easily and quickly orient themselves and then plan their preparation.

Problems

Lecturers have the opportunity to add sample probelms. Like materials, the problems are again divided into categories and marked with appropriate labels. When adding a problem, it is also possible to attach sample solution files, solution explanations, and task tests to the problem condition.

Archive

Ability to add backup information divided into two main types. The first possible type is an archive of previous events. The second possible type is a link to an external site where appropriate archival materials are available.

News

Ability to add news related to upcoming events and programming events. The aim is students and students who are preparing to be informed as soon as possible about these events.

Training

An opportunity for contestants to upload solutions to the sample problems and, after performing the tests, to get information on whether their solution is correct or not.

Searching

Ability to search among the materials and problems added to the system. Lists of materials and problems in the different categories. Lists of materials and problems marked with certain labels.

Administration of the system

The administrator registers users with different roles - contestants, lecturers and administrators, activates user accounts, has access to a list of active users. Other actions with the system that the administrator performs is the management of categories in which the uploaded materials and tasks are separated, and the management of labels to mark different materials and tasks.

BASIC USE CASES OF THE SYSTEM

Basic use cases are represented by the actions that identified user roles can perform with the information objects described.

Basic use cases with information object Materials

User role: Competitor

Each competitor should see a list of all the materials added to the system and a separate list of all the materials added in a particular category. In addition, each contestant must be able to view specific material added and take a closer look at the content. In addition, a competitor must be able to search for a material added to the system.

User role: Lecturer

Lecturers should be able to add materials to prepare the contestants. Each material should be assigned to one or several subcategories to make the information systematized and to facilitate the contestants. Each lecturer should be able to edit the already added **by him/her** training materials. Editing must have the ability to change the title, content, category, and attachments to the material.

Each lecturer should be able to erase material **he/she** adds when he/she thinks that he/she will no longer be of benefit to the contestants.

User role: Administrator

The administrator should be able to edit any added material. Editing must have the ability to change the title, content, category, and attachments to the material.

The administrator should be able to delete any added material when he thinks he will no longer be of benefit to the contestants.

Characteristics related to the categories in which the materials are grouped

User role: Competitor

Every competitor must be able to see the list of all categories added to the system.

User role: Lecturer

Each lecturer should be able to add a new category, edit a category **he/she** created by changing his/her title, add a subcategory to an existing category in the system.

User role: Administrator

The administrator should be able to edit each of the categories added to the system by changing its title.

Basic use cases with information objectArchive

User role: Competitor

Every contestant must be able to see a list of all the archives added. For each archive on the list, there must be title information and a hyperlink to the corresponding online archive.

User role: Lecturer

Each lecturer should be able to add archives.

Each lecturer should be able to edit an archive he has uploaded.

Each lecturer should be able to delete the archives added by him regardless of their type.

User role: Administrator

The administrator should be able to edit any added archive.

The administrator must be able to delete any archive added regardless of its type.

Basic use cases with information objectProblems

User role: Competitor

Every competitor must be able to view a list of all tasks.

Each competitor must be able to review the information related to each task, which includes a detailed description of all attachments separately - type, size. It must be downloadable and must be able to test its own solution for each task.

User role: Lecturer

Each lecturer should be able to add tasks to help prepare contestants for competitions. The environment must provide tools that can be edited by an editor with basic word processing functions to describe the task. Each task must be assigned to one or more subcategories.

Each lecturer should be able to edit the tasks he has already added. This includes:

- Change the title and description
- Upload new files that replace the already uploaded files.

Each lecturer must be able to delete the tasks he has already added. When deleting a task, the files attached to it must remain in the system if used by other objects.

Each lecturer should be able to see the statistics of the decisions of a given task in order to have data on the complexity of the tasks, the different decisions and the users who have tried to solve it. It also has to see how many users have tried to solve the problem, how many experiences each user has done and whether they are successful. It should be possible to view a list of each user's attempts and download the solution from each attempt, whether successful or not.

User role: Administrator

The administrator should be able to perform all possible operations on all tasks in the system: edit and delete all tasks in the system, no matter who their author; When deleting a task, files attached to it must remain in the system if used by other objects.

BASIC USE CASES WITH INFORMATION OBJECT TRAINING

User role: Competitor

Each competitor must be able to upload his / her decision for a task and the system to check it against the uploaded tests for that task. Upload options must include Java source file and C ++ source file. You must be able to choose a compiler. Once the decision has been sent to the system, the competitor has to get information about the task's execution in his account. If the task is successfully completed, the status of the task is changed for this competitor.

User role: System

The system must implement the decisions made by the contestants.

- The system can implement decisions in a consistent manner.
- Any solution received ranks in a queue for execution.
- When more than one solution is received by one user within one minute, only the first one is placed in the execution queue, and the others are ignored.
- After a solution has been executed, the system records the result and sends it to the competitor's profile.
- The system must return to the competitor different statuses according to the execution of the assigned task assignment.
- The possible status returned from the execution of the solution are:

Error	**Significance**
Compilation error	Compilation error of the submitted solution.
Runtime error	Runtime Error (Exception or other Error)
Time limit exceeded	The execution time of the decision is higher than the maximum set by the program author.
Wrong answer	The result of the decision is different from the result set for the tests.
Output format error	The result of the decision is the same as the one expected, but it is in a different format.

Users of the racing platform will have access to it through a web browser and Internet connectivity. This way, it can be used by any device connected to the Internet and having a web browser.

CONCLUSIONS

The following uses have the following benefits for the user:

- Ability to access systemized materials and preparation tasks.
- Ability to send solutions to added tasks.
- Implement the uploaded solutions and compare the results obtained with the expected results.
- Access from different devices and at different times.

The environment project draft for training and preparation for programming competitions can be used as a description of the basic functional requirements for such type of environments. It can be upgraded and developed in favor of its users - programming contestants and / or lecturers who prepare students / students to participate in forums of this type.

REFERENCES

[1] https://www.domjudge.org/. – a Programming Contest System.

[2] http://www.informatika.bg/lectures/contests-and-training-sites- a site for programming contests and training.

[3] https://www.hackerrank.com/– a Programming Contest System.

[4] http://mendo.mk/Welcome.do - System for competitions in informatics.

CONTACT ADDRESS

Pr. Assist. Galina Atanasova, PhD
Department of Informatics and Information Technologies
Faculty of Natural Sciences and Education
Angel Kanchev University of Ruse
8 Studentska Str., 7017 Ruse, Bulgaria
Phone: (+359 82) 888 326
E-mail: gatanasova@ami.uni-ruse.bg

ОСНОВНИ СЛУЧАИ НА УПОТРЕБА В СРЕДА ЗА СЪСТЕЗАТЕЛНО ПРОГРАМИРАНЕ

Галина Атанасова

Русенски университе "Ангел Кънчев"

Резюме: *Статията представя мястото и ролята на системите за обучение при подготовката на ученици и/или студенти за участия в състезания по програмиране. Акцентът е поставен върху проект на система, предназначена за подготовка/самоподготовка за състезания по програмиране. Посочени са основните потребителски роли и свързаните с тях случаи на употреба в такава система. Описаните случаи на употреба могат да се използват като основен набор при проектирането и разработката и на други системи за състезателно програмиране.*

Ключови думи: *Състезания по Програмиране,Системи за състезателно програмиране, Случаи на употреба.*

Requirements and guidelines for the authors - "Proceedings of the Union of Scientists - Ruse"

The Editorial Board of "Proceedings of the Union of Scientists - Ruse" accepts for publication annually both scientific, applied research and methodology papers, as well as announcements, reviews, information materials, adds. No honoraria are paid.

The paper scripts submitted to the Board should answer the following requirements:

1. Papers submitted in Bulgarian, Russian and English are accepted. Their volume should not exceed 8 pages, formatted following the requirements, including reference, tables, figures and abstract.

2. The text should be computer generated (MS Word 97 for Windows or higher versions up to Word 2003) and printed in one copy, possibly on laser printer and on one side of the page. Together with the printed copy the author should submit a disk (or send an e-mail copy to: desi@ami.uni-ruse.bg).

3. Compulsory requirements on formatting:

font - Ariel 12;
paper Size - A4;
page Setup - Top: 20 mm, Bottom: 15 mm, Left: 20 mm, Right: 20mm;
Format/Paragraph/Line spacing - Single;
Format/Paragraph/Special: First Line, By: 1 cm;
Leave a blank line under Header - Font Size 14;
Title should be short, no abbreviations, no formulas or special symbols - Font Size 14, centered, Bold, All Caps;
One blank line - Font Size 14;
Name and surname of author(s) - Font Size: 12, centered, Bold;
One blank line - Font Size 12;
Name of place of work - Font Size: 12, centered;
One blank line;
abstract – no formulas - Font Size 10, Italic, 5-6 lines ;
keywords - Font Size 10, Italic, 1-2 lines;
one blank line;
text - Font Size 12, Justify;
references;
contact address - three names of the author(s) scientific title and degree, place of work, telephone number, Email - in the language of the paper.

4. At the end of the paper the authors should write:

The title of the paper;
Name and surname of the author(s);
abstract; keywords.

Note: If the main text is in Bulgarian or Russian, parts in item 4 should be in English. If the main text is in English, they should be in Bulgarian and have to be formatted as in the beginning of the paper.

5. All mathematical signs and other special symbols should be written clearly and legibly so as to avoid ambiguity when read. All formulas, cited in the text, should be numbered on the right.

6. Figures (black and white), made with some of the widespread software, should be integrated in the text.

7. Tables should have numbers and titles above them, centered right.

8. Reference sources cited in the text should be marked by a number in square brackets.

9. Only titles cited in the text should be included in the references, their numbers put in square brackets. The reference items should be arranged in alphabetical order, using the surname of the first author, and written following the standard. If the main text is in Bulgarian or Russian, the titles in Cyrillic come before those in Latin. If the main text is in English, the titles in Latin come before those in Cyrillic. The paper cited should have: for the first author – surname and first name initial; for the second and other authors – first name initial and surname; title of the paper; name of the publishing source; number of volume (in Arabic figures); year; first and last page number of the paper. For a book cited the following must be marked: author(s) – surname and initials, title, city, publishing house, year of publication.

10. **The author(s) and the reviewer, chosen by the Editorial Board, are responsible for the contents of the materials submitted**.

Important for readers, companies and organizations

1. Authors, who are not members of the Union of Scientists - Ruse, should pay for publishing of materials.

2. Advertising and information materials of group members of the Union of Scientists – Ruse are published free of charge.

3. Advertising and information materials of companies and organizations are charged on negotiable (current) prices.

Editorial Board

ISSN 1314-3077

9 771314 307000