Learning programming by means of composing generalized tasks with random choice by pupils

Pavel S. Pankov *(pps50@rambler.ru)*,

Jyldyz R. Janalieva *(Noledi@yandex.ru)*

(Kyrgyzstan)

Pattaya - 2011

To implement common demands to tests (*Validity, Objectivity* and *Reliability*) and to improve efficiency of testing we offered the following principles:

Generativity. A complete task must not exist before the testing and must be generated (randomly) just before it.

Uniqueness. All examinees must obtain different versions of tasks.

Full confidence. If the testing is official and is conducted by a computer program (generating tasks) then nobody must know answers before finishing.

Concreteness. An answer to a task must be concrete (number, word, short phrase, action).

Definition 1. An algorithm generating different logically correct and methodically proper tasks by initial data (randomly) chosen from finite but sufficiently large sets (ranges) is said to be an *extended task.*

Definition 2. An algorithm permitting (for a teacher) to choose subsets of sets of initial data and generating different logically correct and methodically proper tasks by initial data (randomly) chosen from these subsets is said to be an *adjustable extended task.*

For non-formal express testing of knowledge we offered

Definition 3. The problem is said to be *intellectual eye measurer* (or *measurable imagery*, intuition) if its condition is strict but the answer may be only approximate or approximate answer is permissible; using any tool (computer, paper, reference book) is forbidden;

in sciences the time to answer is about 20 - 30 seconds to avoid immediate counting in head.

If the pupil's answer differs from the exact one less than 10% then the mark is "excellent"; if the error is between 10% and 20% then the mark is "good" etc.

We purposely describe extended tasks not in any algorithmic language but non-formally. Let pupils themselves program tasks offer them to their mates.

All numbers as initial data are random, integer and positive.

Example 1 (simple algebra). "Fill in the blank". Choose *A, B, C* and *D:= A*B+C.* Choose one of three random types of tasks and compose the task 1) *7*.. + 5 = 19* 2) *26 - 6*.. = 14* 3) *16 = 5*.. - 4 (A, C* and *D* are shown; *B* is the answer).

Example 2 (text task in arithmetic).   Random identity: 280124+40*45/60=280154 and three random versions of text with the same numbers:

At 7.30 the speedometer showed 280124 miles,…

1) at 8.15 it did 280154 miles. What was the average velocity (mph)?

2) the average velocity is 40 mph. What will it show at 8.15?

3) the average velocity is 40 mph. When will it show 280154 miles?

Example 3 (understanding of "Area"). Show a unit square and a random figure of area about 5..10 units. (The program can find the area of the figure by counting pixels of such color).

Task 3a. Input the area of the figure (approximately).

Task 3b. Enlarge the square by pulling its right upper corner until its area is equal the area of the figure (approximately).

Example 4 (measuring imagery in geography).

Estimate the distance between two random (pointwise) objects (cities, well-known peaks, capes). Here the list of objects (some hundreds) is too vast to learn all mutual distances by heart (the program counts answers by means of formulas of spherical trigonometry).

But if a pupil can image the map (the globe) then s/he is able to obtain the excellent mark.

Thus, a pupil cannot learn the answer by heart and s\he is to learn the subject thoroughly.

Conclusion

We hope that each pupil would be able to choose a generalized task of suitable complexity and to implement it with the teacher's assistance as an interesting and useful program for other pupils.

Thank you for attention!