DICE Coding Training Program

Overview

Goal: Make candidates for positions at DICE fit for their target position (mainly regarding their programming skills)

Advantages for the candidate

The candidate can train and improve his/her programming skills based on real-world programming tasks. Every task will have a detailed explanation and a responsible person that can be contacted if questions arise. It will also include feedback from experienced programmers and researchers with which the candidate can learn from errors. Additionally, it is a good way for the candidate to stay in contact with the DICE group and present him/herself over a longer time.

Advantages for the DICE group

Simple programming tasks can be implemented by candidates. Additionally, it gives the opportunity to stay in contact with candidates and learn more about their abilities.

Responsibilities of DICE members

- Identify tasks that are easy to implement without having too much detailed knowledge about the project itself
  - Write a description for the tasks that enables people new to the field to understand it
- Answer questions of candidates. Note that this includes the possibility to terminate the collaboration on a particular task if it goes far beyond the abilities of the candidate.
- Give feedback to solutions, e.g., via the review function of github.

Workflow

(D) stands for a DICE member while (C) stands for the candidate.

1. (D) identifies tasks that are easy to implement
   1. Create an issue for each task on GitHub. Each task should have a detailed description and a contact person.
   2. Mark the issue with a “training” label.
2. (C) solves the tasks.
   1. He/she regularly checks **here**
   2. For interesting tasks, (C) forks the project and implements the solution on his/her fork.
   3. After the solution has been implemented and tested, (C) creates a pull request to apply the changes to the original project. (The description might contain the information into which branch the pull request should merge the changes)
3. (D) reviews the changes (e.g., using the review mechanism of Github) and gives feedback.
4. (C) applies changes if (D) asked for them.
5. Repeat step 3 and 4 until convergence.