

Produção de Concept Inventory a partir do estudo da linguagem de programação Java

Renan Paes Leme de Souza*, Ricardo Edgard Caceffo

Resumo

A partir do estudo de erros frequentes na lógica de programação (Misconceptions) das linguagens C e Python, desenvolveram-se códigos na linguagem Java de forma a detectar novos Misconceptions e a verificar se os encontrados nas linguagens C e Python também ocorrem em Java. Finalmente, elaboraram-se questões a fim de montar um Concept Inventory, um questionário que ajudará a detectar potenciais Misconceptions cometidos por alunos de cursos introdutórios e a reforçar o aprendizado, evitando ao máximo a incidência desses erros.

Palavras-chave:

Java, Concept Inventory, Misconception

Introdução

Um dos maiores desafios em cursos introdutórios de programação é desenvolver uma metodologia que permita maximizar o aprendizado dos alunos, bem como ensiná-los a refletir sobre o desenvolvimento da lógica de programação.

Uma proposta de solução consiste na aplicação de um *Concept Inventory*, um questionário desenvolvido a partir de programas computacionais e que contém, em suas alternativas incorretas, os erros mais frequentes na linguagem de programação lecionada, denominados *Misconceptions*.

Dessa forma, pode-se detectar de maneira eficiente as principais deficiências de aprendizado em cada turma e auxiliar o docente a reforçar determinados conceitos de uma linguagem de programação.

Resultados e Discussão

Inicialmente, realizou-se uma análise¹ dos *Misconceptions* encontrados nas linguagens C e Python e, a partir do desenvolvimento de códigos em Java, procurou-se verificar se as ocorrências desses erros também eram válidas para essa linguagem de programação.

Na sequência, esses códigos foram utilizados como base para detectar potenciais *Misconceptions* exclusivos da linguagem de programação Java, especialmente em aplicações do paradigma de orientação a objetos. Ao término dessa etapa, os 33 *Misconceptions* detectados foram divididos em 8 categorias, de acordo com o tópico de lógica de programação envolvido.

O próximo passo foi construir a documentação para cada um dos *Misconceptions*. Para isso, usou-se a estrutura de um *Antipattern*¹, um conceito desenvolvido para exemplificar erros e más práticas de programação, que não devem ser utilizadas. Para o projeto, documentou-se o tipo de erro, um trecho de código exemplificando-o, as possíveis razões de um aluno cometê-lo, as consequências em um programa computacional e sugestões para evitar que esse erro seja cometido.

Finalmente, após a construção da documentação, iniciou-se a produção de um *Concept Inventory*, um questionário contendo perguntas de múltipla escolha que visam detectar equívocos conceituais de estudantes em diferentes tópicos de lógica de programação. Assim, foram elaboradas questões cujas alternativas incorretas contêm respostas obtidas quando um estudante comete um erro caracterizado por um *Misconception*. Com as

questões finalizadas, realizou-se uma seleção das mais relevantes para a linguagem de programação Java e, com as escolhidas, construiu-se o *Concept Inventory*. Apesar de não ter sido validado por um grupo de estudantes e um docente, espera-se que ele apresente bons resultados nos cursos introdutórios de programação.

Conclusões

Após a obtenção dos *Misconceptions* e a construção do *Concept Inventory*, pode-se concluir que os resultados foram satisfatórios, próximos do esperado inicialmente.

Ao comparar os resultados obtidos nas linguagens C e Python, percebe-se que a incidência desses *Misconceptions* na linguagem Java é elevada, o que indica uma considerável correspondência entre as três linguagens. Apesar de não existirem ocorrências diretamente ligadas ao tópico de ponteiros, a linguagem Java possui *Misconceptions* ligados ao paradigma de orientação a objetos. Logo, seus conceitos, ainda que de maneira introdutória, devem ser ensinados pelos docentes.

Quanto ao *Concept Inventory*, espera-se que ele tenha grande importância no reforço do aprendizado dos alunos. Apesar de sua validação não ter sido realizada, a elevada correspondência entre as linguagens C, Python e Java indica que ele envolverá os principais tópicos de Java, com potencial para o aparecimento de erros frequentes na lógica de programação dos alunos, ou seja, *Misconceptions*.

Agradecimentos

Agradecimentos ao Instituto de Computação (IC) da Unicamp e à agência de fomento à pesquisa PIBIC pelo apoio e incentivo ao desenvolvimento desse projeto.

¹ CACEFFO, R. FRANÇA, B.; GAMA, G.; BENATTI, R.; APARECIDA, T.; CALDAS, T.; AZEVEDO, R. (2017) An Antipattern Documentation about Misconceptions related to an Introductory Programming Course in C. In Technical Report 17-15, Institute of Computing, University of Campinas, SP, Brasil. 42 pages. October, 2017.