Using a Low-Code Environment to Teach Programming in the Era of LLMs

JETBRAINS Research

Anna Potriasaeva [1], **Katsiaryna Dzialets** [2], **Yaroslav Golubev** [1], **Anastasiia Birillo** [1]

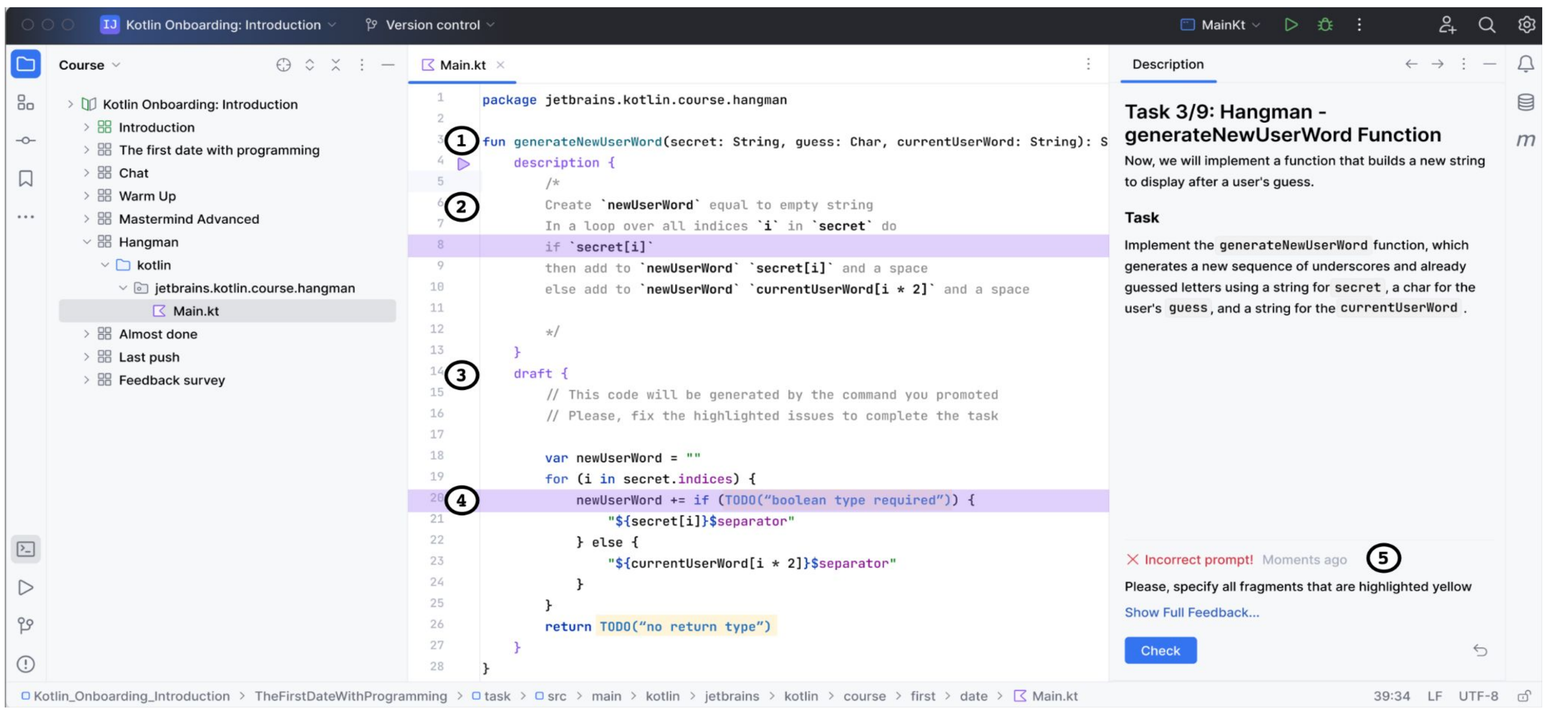
JetBrains Research[1], JetBrains[2]

Summary

LLMs change the landscape of software engineering, and the question arises: **How can we combine LLMs with traditional teaching approaches in computer science?** In this work, we propose to teach students in a *low-code environment*, developing not only their coding but also *problem decomposition* and *prompting* skills.

Approach

Research shows that LLMs are crucial for teaching students both higher-level concepts and the practical skills of prompting that are becoming increasingly important.[1, 2, 3, 4] The **key idea** of our approach is to use *intelligent prompt engineering* to teach algorithmic thinking and problem decomposition, while combining it with code generation and direct coding.



(1) The students learn the basics of problem decomposition so that they can work with the program at the level of functions.
(2) For the given task, the student describes the necessary algorithm in natural language using intelligent prompt engineering, meaning that the prompt may contain code if the student is already familiar with some programming concepts.
(3) The student can `RUN` a particular `description` block to see the current version of the generated code in the `draft` section and the errors in it.
(4) The errors are highlighted in both blocks. In our vision, the student can only fix errors in the `description`, but we are considering making it possible in the `draft` section too.
(5) The student can `CHECK` their solution, generating all the code and launching the task's tests.

Current state: we developed the first prototype that covers all the proposed ideas and can be evaluated with the students.

[1] *Decker, Adrienne, et al. "Piecing together the next 15 years of computing education research workshop report.", 2022.*
[2] *Denny, Paul, et al. "Prompt Problems: A new programming exercise for the generative AI era.", 2024.*
[3] *Tedre, Matti, and Henriikka Vartiainen. "K-12 computing education for the AI era: From data literacy to data agency.", 2023.*
[4] *Prather, James, et al. "The robots are here: Navigating the generative ai revolution in computing education.", 2023.*

Technical details

General details:
- Supports Kotlin language
- Integrates into the in-IDE learning format of the JetBrains Academy plugin

Under-the-hood:
- Uses a DSL to support `description` and `draft` blocks
- Uses a grammar to teach the students to write concrete prompts
- Uses static analysis to:
  - check whether students only use defined variables and functions;
  - analyze the model's output, improve its code quality, and put `TODO` statements where needed

Possible evaluation

Experiment design:
- Testing with a group of first-year Bachelor students
- Solving the same course in a traditional and a new ways
- Comparing the results on the set of control tasks
- Have an interview with teachers about their perceptions

Research questions:
- What are the students' perceptions of the proposed approach?
- How does the proposed approach affect code solution's quality?
- What are educators' perceptions regarding the effectiveness and utility of the proposed approach?

Open questions

- How can we measure the impact of the proposed approach on the learning process?
- Is it possible to extend this approach to more complex tasks and large projects?
- Can this approach be applied to non-computer science students to help them learn to code?
- How can the traditional education be transformed in the new era of the LLMs in programming?

Future work

- Conduct a study with students and evaluate the learning effect of the proposed approach. Check if students are better at solving programming problems using this approach.
- Define the scope of this approach: should students use this approach only at the beginning of learning to code, or should they keep this methodology as the main way of programming in general?

Links

Full text

Demo-video

JetBrains Academy plugin

Contacts

Anna Potriasaeva anna.potriasaeva@jetbrains.com
Katsiaryna Dzialets katsiaryna.dzialets@jetbrains.com
Yaroslav Golubev yaroslav.golubev@jetbrains.com
Anastasiia Birillo anastasia.birillo@jetbrains.com