

Система анализа поведения студентов при обучении программированию

Люлина Елена Сергеевна

научный руководитель: доц. кафедры СП,
к.т.н. Т.А. Брыксин

рецензент: генеральный директор ООО “Цифровые
образовательные решения” Н. И. Вяххи

СПБГУ, Математическое обеспечение и
администрирование информационных систем,
Системное программирование

6 июня 2020 г.

Зачем это надо?

Самостоятельное решение задач вызывает трудности у студентов¹

Понимание поведения и ошибок студентов поможет экономить ресурсы преподавателей

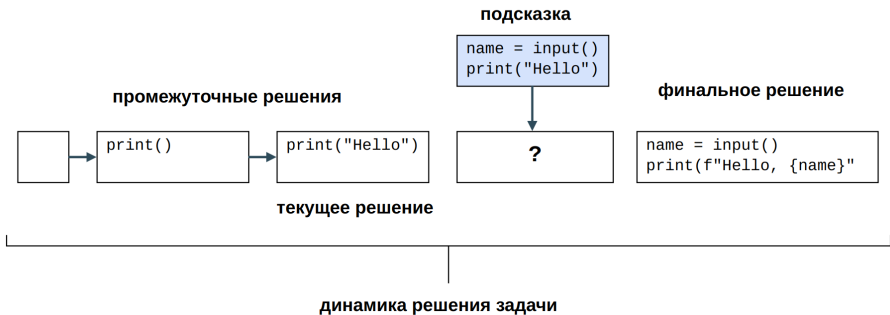
Данные о решениях студентов помогут автоматизировать процесс подсказок

Для эффективности онлайн-курсов и самостоятельного обучения

¹ Lahtinen Essi, Ala-Mutka Kirsti, Järvinen Hannu-Matti. A Study of the Difficulties of Novice Programmers // ITiCSE'05. -- 2005.

- Разработка алгоритма генерации подсказок для решения учебных задач на языке Python

- сделать обзор существующих решений
собрать датасет с динамиками решений задач
разработать алгоритм генерации подсказок
протестировать алгоритм и оценить релевантность подсказок



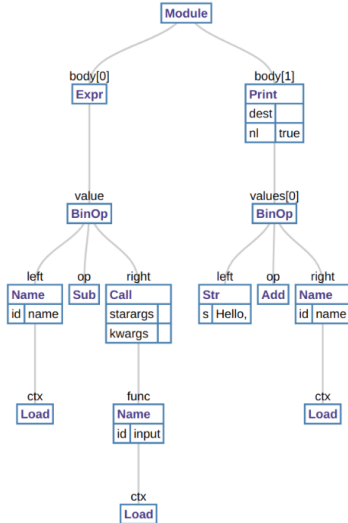
Фрагмент кода

```
name = input()
print("Hello, " + name)
```

Изменения AST:

1. удаление вершины
2. добавление вершины
3. замена вершины

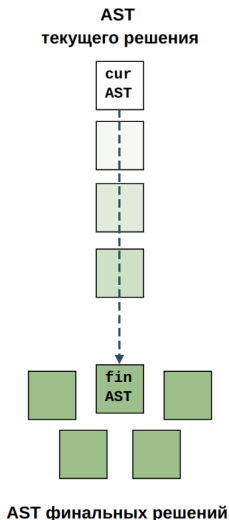
Abstract Syntax Tree (AST)



Генерация подсказок для языка Python²:

предлагается алгоритм
каноникализации AST
используются только
финальные решения
подсказка ищется
перебором булеана
изменений
плохо работает на
сложных задачах

² Rivers, K. "Data-Driven Hint Generation in Vast Solution Spaces: a Self-Improving Python Programming Tutor", International Journal of Artificial Intelligence in Education — 2017.

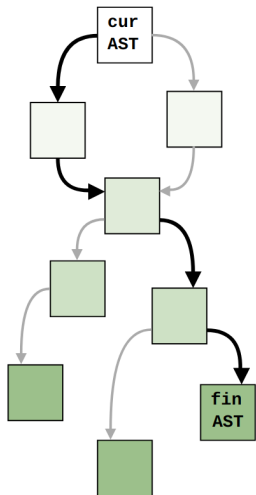


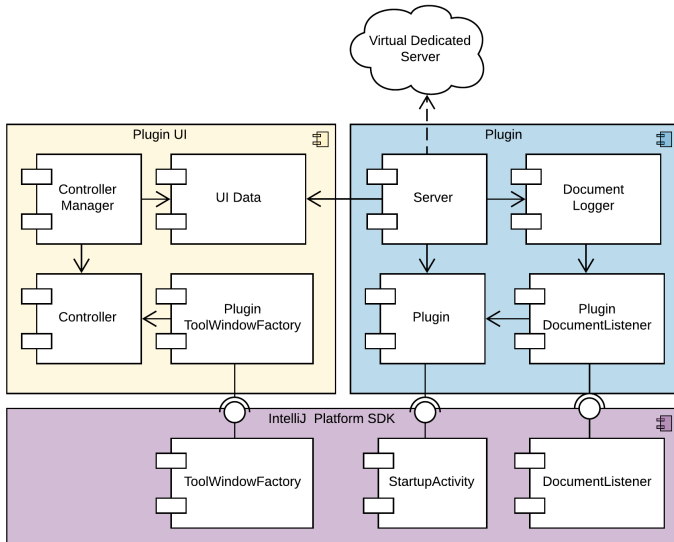
Генерация подсказок для блочных языков³:

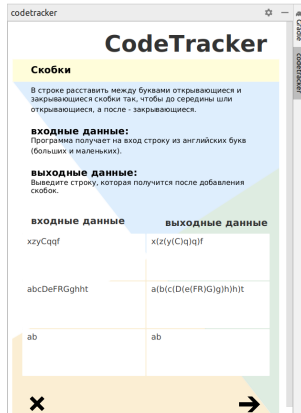
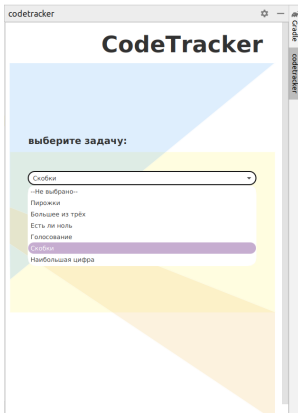
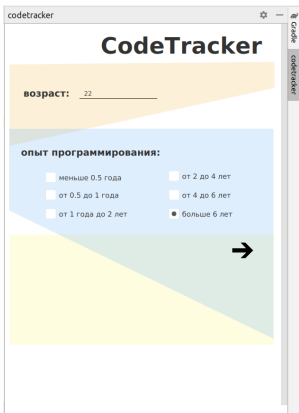
используются динамики решений, объединенные в пространство решений
подсказка ищется как следующее решение на пути к финальному
требуется дополнительных данных

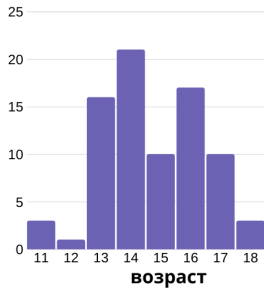
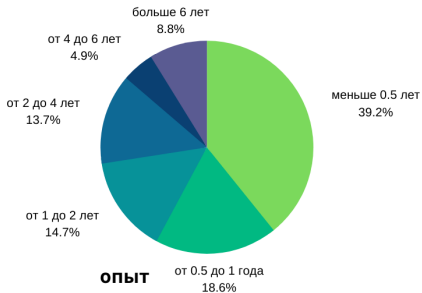
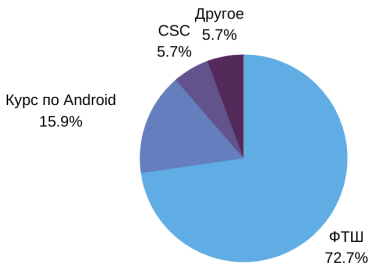
³ Piech, C. "Autonomously Generating Hints by Inferring Problem Solving Policies", L@S '15 — 2015.

пространство решений

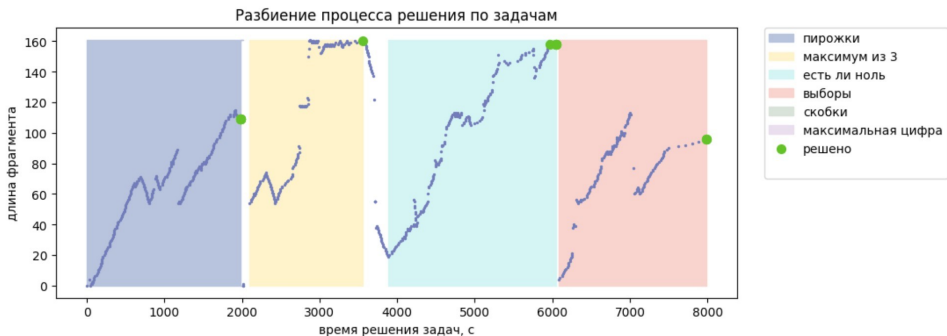








фильтрация некорректного кода
оценка решений задач при помощи тестов
очистка кода от лишних фрагментов



Основные особенности алгоритма:

использует пространство решений для текстовых программ

позволяет применять алгоритмы генерации подсказок для блочных языков

использует алгоритм каноникализации для сжатия пространства решения

использует более точный подсчёт изменений между фрагментами кода

учитывает опыт и возраст

учитывает структуру фрагмента

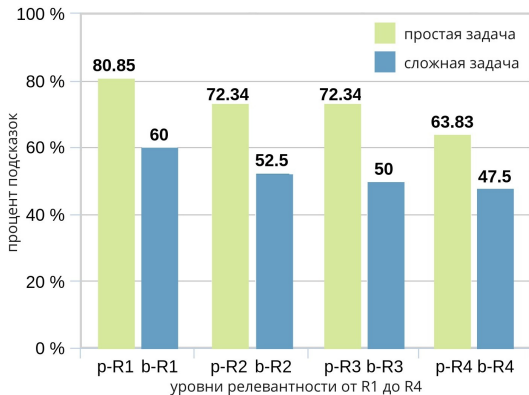
. сколько будут стоить n пирожков, если один пирожок стоит a рублей и b копеек?

<p>1. aAST текущего фрагмента:</p> <pre>g4 = int(input()) g3 = int(input()) g2 = int(input()) g0 = (g4 * g2) g1 = (g3 * g2) if (g1 >= 100): g5 = (g1 // 100) g1 = (g1 - (g5 * 100)) g0 = (g0 + g5) print(g0, g1)</pre>	<p>aAST, предложенное в качестве подсказки:</p> <pre>g4 = int(input()) g3 = int(input()) g2 = int(input()) g0 = (g4 * g2) g1 = (g3 * g2) if (g1 >= 100): g5 = (g1 // 100) g0 = (g0 + g5) g6 = (g1 % 100) g1 = g6 print(g0, g1)</pre>
<p>2. aAST текущего фрагмента:</p> <pre>g4 = int(10) g3 = int(15) g2 = int(2) g1 = (g2 * g4) g0 = (g2 * g3) print()</pre>	<p>aAST, предложенное в качестве подсказки :</p> <pre>g4 = int(input()) g3 = int(input()) g2 = int(input()) g1 = (g2 * g4) g0 = (g2 * g3)</pre>

критерии подсказки:

- логичность
- шаг
- структура
- приближение к решению

оценены 50 подсказок
для двух задач
различной сложности
определены 4 уровня
релевантности



сделан обзор существующих решений
создан инструмент для сбора данных
собран набор данных, включающий динамики решений
реализованы алгоритмы обработки данных, построения пространства решений
разработан алгоритм генерации подсказок с использованием пространства решения
протестирован алгоритм генерации подсказок для задач разного уровня сложности, достигнуты 80.85% и 60% релевантных подсказок

