Никита Босов

Тема работы: расширяемый генератор синтаксически корректных программ для обучения программированию

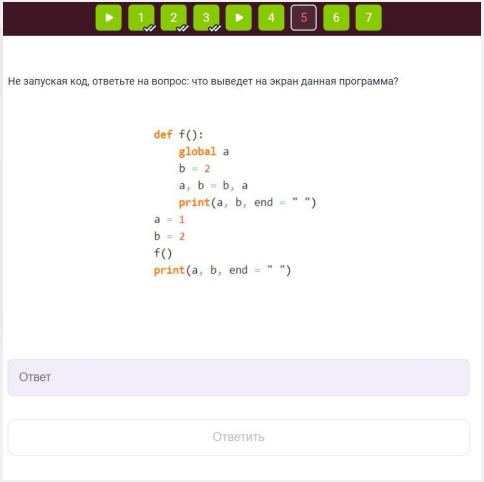
Выпускная квалификационная работа

Научный руководитель: М. М. Заславский

15 июня 2022

Факультет математики и компьютерных наук СПбГУ
Программа «Современное программирование»

Задачи для курсов по языкам программирования



МКН

Генераторы программ

- Генератор программ — программа, генерирующая программный код согласно некоторым условиям.
- Генераторы программ используются в основном для тестирования компиляторов и интерпретаторов.

Аналоги

- Программы, используемые для тестирования компиляторов: Csmith, Yarpgen.
- Deepsmith — инструмент генерации кода на OpenCL, использующий машинное обучение.

Цель — разработка генератора программ, позволяющего создавать примеры кода (в виде текста или изображения) для задач курсов обучения языкам программирования.

Задачи:

1. Разработать архитектуру генератора программ.
2. Разработать и описать способ задания основной логики программ.
3. Создать интерфейс для добавления поддержки генерации кода на разных ЯП.
4. Поддержать возможность проверки ответов студентов.

Архитектура проекта и используемые технологии

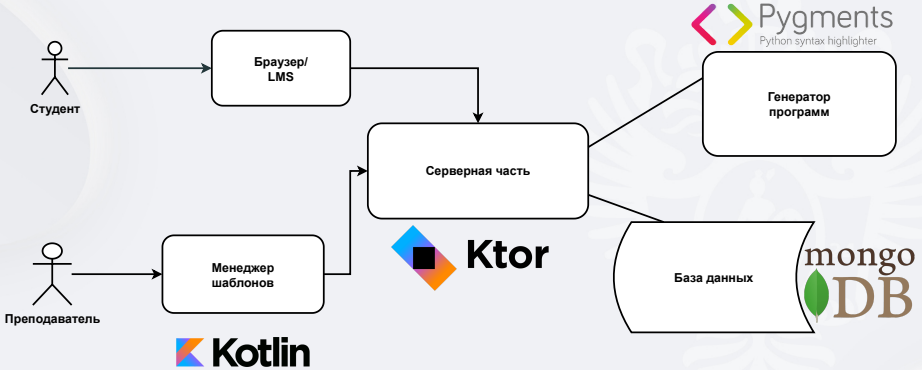
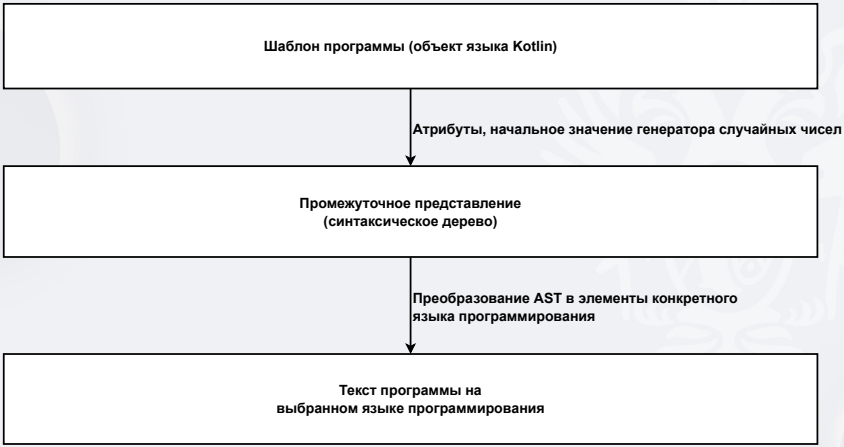


Схема генерации программ



Пример генерации

Шаблон программы на Python:

```
fun task(): ProgramTemplate<out ProgramLanguageTag> = ProgramTemplate<PythonTag> {
    mainFun {
        val xVar = variable("x")
        val stringLen = randomNumConstant(10, 20)
        val randomString = randomStringConstant(stringLen)
        addVarDef(xVar, randomString)
        `if`(funcCall("len", xVar) lt constant(15)) {
            addFuncCall("print", constant("small"))
        }.`else` {
            addFuncCall("print", constant("long"))
        }
    }
}
```

МКН

Менеджер шаблонов

```
Usage: task-manager options_list
Arguments:
    Command { Value should be one of [add, delete] }
Options:
    --host [0.0.0.0] -> Host of code generator { String }
    --port [8080] -> Port of code generator { Int }
    --help, -h -> Usage info
```

МКН

Пример генерации

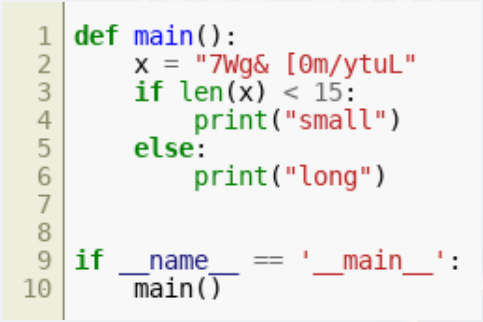
Сгенерированная программа на Python:

Текст программы

```
def main():
    x = "7Wg& [0m/ytuL"
    if len(x) < 15:
        print("small")
    else:
        print("long")


if __name__ == '__main__':
    main()
```

Изображение



Пример генерации

Шаблон программы на C++:

```
fun task(): ProgramTemplate<out ProgramLanguageTag> = ProgramTemplate<CppTag> {
    mainFun {
        val xVar = variable("x")
        val stringLen = randomNumConstant(10, 20)
        val randomString = randomStringConstant(stringLen)
        addVarDef(xVar, "string", randomString)
        `if`(funcCall("size", xVar) lt constant(15)) {
            addFuncCall("printf", constant("small"))
        }.`else` {
            addFuncCall("printf", constant("long"))
        }
    }
}
```

МКН

Пример генерации

Сгенерированная программа на C++:

Текст программы

```
#include <bits/stdc++.h>
using namespace std;
int main(int argc, char **argv) {
  string x = " _sMYL> kv";

  if (size(x) < 15) {
    printf("small");
  } else {
    printf("long");
  }
}
```

Изображение

Проверка ответов

- Запуск программы и сохранение ее вывода происходит на этапе создания кода и изображения.
- Сравнение ответов — построчно
- Результат проверки — процент правильных строк в ответе.

Дальнейшее развитие

- Поддержка новых элементов синтаксиса
- Улучшение синтаксиса шаблонов.
- Поддержка генерации кода на других языках программирования.
- Улучшение качества генерируемых изображений.
- Интеграция в образовательные платформы (Moodle).

Результаты работы

Создано приложение для генерации примеров кода для учебных задач по шаблону и параметрам с поддержкой проверки ответов. Данную систему можно расширять для поддержки других языков программирования.

Никита Босов, neckbosov@gmail.com,
https://github.com/OSLL/bsc_bosov

Поддержка генерации кода на разных ЯП

```
interface CodeMapper<LanguageTag : ProgramLanguageTag> {
    fun generateCode(program: Program<in LanguageTag>): String
}
```

```
{
  "python": {
    "extension": "py",
    "formatCmd": "autopep8 -i $PROGRAM_PATH",
    "compileCmd": null,
    "runCmd": "python3 $PROGRAM_PATH"
  },
  "cpp": {
    "extension": "cpp",
    "formatCmd": "clang-format -i $PROGRAM_PATH",
    "compileCmd": "g++ -std=c++20 -o $EXE_PATH $PROGRAM_PATH",
    "runCmd": "$EXE_PATH"
  }
}
```

МКН