

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»

# Цель дипломной работы

- На сегодняшний день существует множество курсов по обучению языкам программирования.
- Одни из самых популярных задач в таких курсах — задачи вида «что выведет данный код».
- Для таких задач на сегодняшний день необходимо писать самому или искать примеры программ.
- Цель проекта — создать генератор таких программ.



# Генераторы программ

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



# Аналоги

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

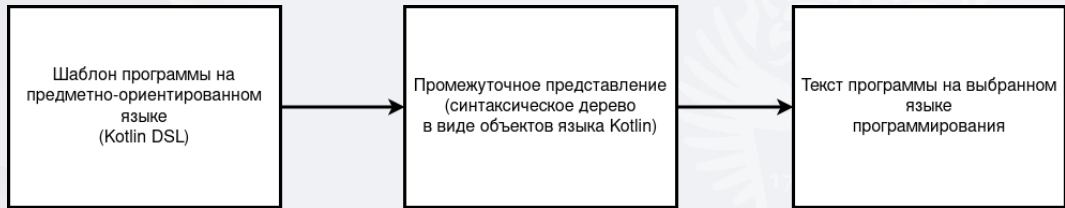


# Постановка задачи

Разработать генератор обучающих примеров программ в виде текста или изображения с поддержкой генерации кода на разных языках программирования и управления задачами, на основе которых генерируется код.



# Схема генерации программ



## Подзадачи

1. Разработка универсального представления сгенерированных программ (IR), которое можно преобразовывать в код на разных языках программирования. Также разработать представление шаблонов, которые можно сохранять в базу данных и преобразовывать в IR при подстановке параметров.
2. Разработка предметно-ориентированного языка построения шаблонов программ.



## Подзадачи

3. Добавить возможность добавления и удаления шаблонов.
4. Добавить возможность генерации изображения программного кода.
5. Добавить возможность проверки ответов студентов.



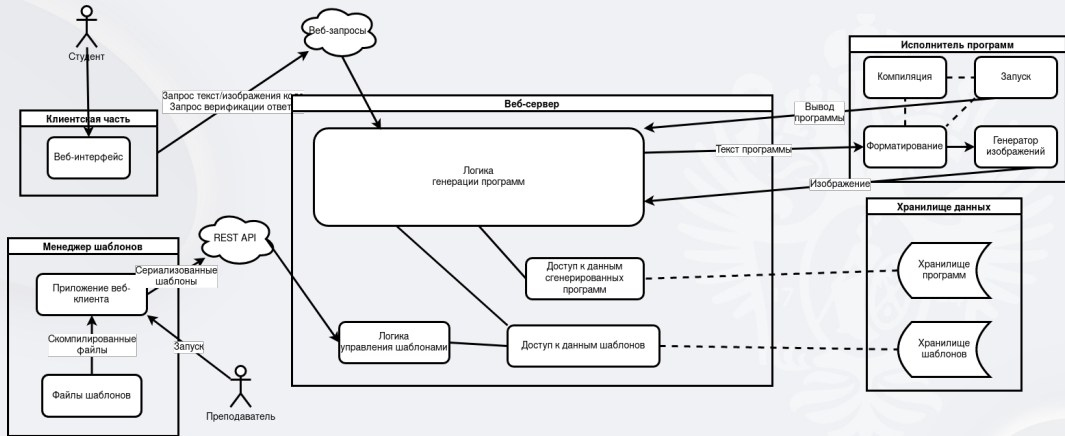


# Архитектура проекта и используемые технологии

- Веб-сервер — фреймворк Ktor
- Преобразование шаблонов кода в JSON — библиотека `Kotlinx.serialization`
- База данных — MongoDB
- Генерация изображений кода — библиотека `python-pygments`



# Схема архитектуры проекта



# Пример генерации

## DSL:

```
val task = ProgramTemplate<PythonTag> {  
    val xVar = variable("x")  
    val stringLen = randomNumConstant(10, 20)  
    val randomString = randomStringConstant(stringLen)  
    addAssignment(xVar, randomString)  
    addIfElseInstr(funcCall("len", xVar) opLt constant(15)) {  
        addFuncCall("print", constant("small"))  
    }.addElse {  
        addFuncCall("print", constant("long"))  
    }  
}
```



# Пример генерации

## Сгенерированная программа:

Текст программы

```
x = ".W,hHm)1^,80^6"  
if len(x) < 15:  
    print("small")  
else:  
    print("long")
```

Изображение

```
1 x = ".W,hHm)1^,80^6"  
2 if len(x) < 15:  
3     print("small")  
4 else:  
5     print("long")
```



# 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
- Поддержка генерации кода на других императивных языках.
- Интеграция в образовательные платформы (Moodle).



# Результаты работы

1. Разработан предметно-ориентированный язык для написания шаблонов программного кода.
2. 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](mailto:neckbosov@gmail.com),  
[https://github.com/OSLL/bsc\\_bosov](https://github.com/OSLL/bsc_bosov)

