

правилам грамматики. В результате успешной работы на выходе получаем дерево разбора. Третий этап реализуется семантическим анализатором для проверки смысловой связи между узлами дерева. В случае возникновения ошибок на любой фазе генерация кода не производится. Заключительным этапом является генерация кода в язык *assembler*.

Особенностями транслятора *ТАО-2021* является:

1. Поддержка разных систем счисления и взаимодействие с ними посредством префиксов и перевода их десятичную систему.
2. Поддержка компилятором классификаций ошибок на разных этапах трансляции языка.
3. Создание отдельного типа входных символов «N» — цифры
4. Легкость использования языка, семантика языка напоминает семантику языка программирования Python, также гибкость языка позволяющие несколько способов определения и инициализации переменных, благодаря большому дереву разбора.
5. Наличие документации о возможных ошибках с примерами, которая позволяет более быстрое погружение языка.

Создана программа автоматической проверки компилятора в виде 50 unittest-тов написанных на языке программирования Python, которая позволяет ускорить дальнейшую разработку и поддержку нового языка программирования на определенном жизненном цикле программы.

УДК 004.415.2

Студ. С.Л. Скалкович, Н.М. Чигоя
Науч. рук. ст. преп. А.С. Наркевич
(кафедра программной инженерии, БГТУ)

РАЗРАБОТКА ПРОЕКТА BSTUDENT HUB

Проект BSTudent Hub это многостраничный сайт с разными модулями – главная страница, бинарные вычисления, модуль развлечений и многофункциональное расписание. Главной задачей проекта являлась применение на практике знаний командной разработке полученных в ходе дисциплины «Технологии разработки программного обеспечения».

Для четкой и эффективной работы команды был выбран смешанный стиль разработки – Scrum и Kanban. Так как эти системы являются частями семейства гибких итеративно-инкрементальных методов управления проектами и продуктами Agile, мы старались следовать основным идеям данной системы.

В ходе разработки проводились собрания, где команда находила решение проблем, обсуждала идеи и предложения, устанавливала дедлайны.

Для удобства разработки, была создана доска «Канбан», где находились карточки с кратким планом к заданию, именами ответственных за выполнение поставленного задания.

Основная разработка велась с помощью системы контроля версий Git и GitHub. Перед началом разработки был создан репозиторий под названием BSTUdent Hub, где находятся ветки: main, styles, INFO, timetable, game, binary-code. В ходе разработки, по достижению пунктов тестирования или подтверждения, ответственный за тестирование и Product Manager заходили в соответствующую ветку и выполняли проверку, в зависимости от успеха проверки карточка с заданием отправлялась на доработку, с проблемами, указанными проверяющими или переходила на следующую стадию.

При достижении пункта «тестирование» в плане карточки с заданием проводилось негативное и позитивное тестирование для каждого блока, тестирование было разделено на две таблицы которые включали в себя:

- 1 входные и выходные данные;
- 2 проверка корректной работы окна модуля.

При обнаружении ошибки или недочёта, в одной из колонок ставилось failed, а в другой прописывалась суть ошибки. После исправления в колонке о выполнении значение изменялось на passed.

По окончании тестирования модуля подготавливался вывод о его работоспособности.

Документация о проекте была размещена в wiki репозитория и отдельной ветке Info. В ней можно найти всю необходимую информацию о создании проекта, технические задания, отчеты о работе модулей и итогах тестирований.

ЛИТЕРАТУРА

1. Scrum, Kanban или оба? [Электронный ресурс] – Режим доступа: <https://habr.com/ru/company/otus/blog/594285/> – Дата обращения 25.11.2021
2. Методология Kanban: введение [Электронный ресурс] – Режим доступа: <https://habr.com/ru/post/230725/> – Дата обращения 22.11.2021
3. Гибкая методология разработки “Scrum” [Электронный ресурс] – Режим доступа: <https://habr.com/ru/post/247319> – Дата обращения 17.11.2021