В настоящее время нами ведется разработка интернет-сообщества для аспирантов, магистров, учителей и преподавателей вузов. Проект “Teachers community” находится по адресу http://teachcom.ru.

Особенность данного проекта состоит в том, что в нем совмещен функционал двух видов социальных сетевых сервисов, социальной сети и вики-ресурса, а также имеются внутренние сетевые сообщества, которые разделены в соответствии с паспортами специальностей ВАК

Зарегистрированные пользователи проекта получают стандартный функционал социальной сети (личные сообщения, друзья, группы, лента событий, форумы обсуждений).

С помощью раздела документы пользователи получают доступ к функционалу вики-ресурсов. Создавая документ, пользователи имеют возможность:

1. Многократно редактировать содержание этого документа.
2. Использовать особый язык разметки при создании и редактировании данного документа.
3. Соавторства.
4. Изменять доступ для определенных пользователей к просмотру, редактированию и комментированию документа.

На странице «Центры совместной работы» имеется 18 основных внутренних сообществ в соответствии с классификацией специальностей ВАК. Это позволяет пользователю сразу после регистрации начать взаимодействие со своими коллегами, которых можно с легкостью найти в нужном внутреннем сообществе.

В качестве вывода, хотелось бы отметить, что объединение функционала самого популярного сетевого сервиса (социальной сети) и мощнейшего сайта-представителя второго поколения Веб (Вики) в одном ресурсе, переводит интерактивное взаимодействие педагогов на новый уровень. Преподаватели смогут осуществлять в рамках одного ресурса, как индивидуальное взаимодействие друг с другом, так и коллективное. Молодые ученые получат возможность непосредственно перенимать педагогический опыт у ведущих ученых из других регионов. В дополнении с классификацией внутренних сообществ в соответствии со специальностями ВАК, делает наш ресурс удобной современной средой для интерактивного взаимодействия педагогов.

Библиографический список

1. *Патаракин Е. Д.* Социальные сервисы Веб 2.0 в помощь учителю/ Е. Д. Патаракин. М.: Интуит.ру, 2007. 64 с.
2. *Семеренко Р.Г.* Актуальные вопросы информационно-педагогической подготовки аспирантов/ Р.Г. Семеренко. М.: ГУУ, 2012. 184-187 с.

С.А. Рудаков

КОНСТРУИРОВАНИЕ ИМЕН И ОФОРМЛЕНИЕ ИСХОДОГО КОДА

rusear@list.ru
Челябинский государственный университет, Челябинск

Below there are provided the convincing arguments that the 'programming style' is not related to the programs design rules; that the requirement to the names construction in accordance to their usage is unreasonable; the programs code readability should be supported by the detailed comments explaining the programmed algorithm; it's unacceptable to require to design the programming code

according to any rules which are not consequent from the programming language syntax; the programming code design rules are the terms of corporate agreement.

При обучении студентов языкам программирования я столкнулся с неожиданной проблемой оформления программ. Многие преподаватели начинали обучение программированию с формирования требований к оформлению программ, среди которых конструирование имен и запись программы "уступами". Соблюдение этих требований как "хороший стиль программирования" было положено в основу выполнения лабораторных работ. Эти требования также выдвигаются и преподавателями в школах как основа обучения программированию. В книге Джесс Либерти и Брэдли Джонс [1] написаны рекомендации: "*Уважающие себя программисты стремятся избегать таких нечитабельных имен переменных, как J23qrsnf, а однобуквенные имена (например, x или i) используют только для временных переменных, таких как счетчики циклов. Старайтесь использовать как можно более информативные имена.*" Далее приведен пример функции, вычисляющей z=x*y и рекомендация заменить безликие переменные на понятные названия, в результате чего формула станет выглядеть понятней Area= Width * Length. Факультет математики, механики и компьютерных наук Южного федерального университета опубликовал указания по оформлению текстов программ на языке Паскаль [2]. В этих указаниях на 8 страницах четко прописаны правила оформления, по которым нельзя писать несколько операторов в одной строке, продолжение записи в другой строке необходимо дополнять двумя пробелами, внутренние блоки необходимо сдвигать на два пробела и т.д.

Некоторые выдержки из книги [3].

"*Создание читаемой программы служит признаком хорошего стиля программирования. Это приводит к облегчению понимания смысла программы, поиска ошибок и в случае необходимости ее модификации....*

Мы уже упоминали о двух таких способах: выбор осмысленных обозначений для переменных и использование комментариев....

Еще один прием состоит в использовании пустых строк для того, чтобы отделить одну часть функции, соответствующую некоторому семантическому понятию, от другой. ... Синтаксические правила языка Си не требуют наличия пустой строки в данном месте, но поскольку это стало уже традицией, то и мы делаем также.

Четвертый принцип, которому мы следуем, заключается в том, чтобы помещать каждый оператор на отдельной строке. Опять же это только соглашение, которое никак не регламентируется правилами языка, так как Си имеет "свободный формат".

Проблема, с которой я столкнулся, является не локальной, а глобальной. В настоящей статье предлагается решение этой проблемы.

Рассмотрим основные понятия.

Парадигма программирования — *это комплекс концепций, принципов и абстракций, определяющих фундаментальный стиль программирования* [4].

Парадигма программирования (стиль программирования) — *это система идей и понятий, определяющих стиль написания компьютерных программ. Это способ концептуализации, определяющий организацию вычислений и структурирование работы, выполняемой компьютером* [5].

Вывод 1. Приведенные энциклопедические понятия показывают, что термин "стиль программирования" никак не связан с правилами оформления программ.

Уже достаточно давно стили программирования задают зарубежные программисты. При программировании они используют читаемые (на английском языке) длинные имена. Примеров множество. Открываем заголовочный файл tchar.h в оболочке Microsoft Visual Studio 2010. Возможно имена __DEFINE_CPP_OVERLOAD_SECURE_FUNC_0_3, _CRT_INSECURE_DEPRECATE, _mbsnbicoll_l говорят о чем-то человеку, свободно владеющему не просто английским языком, а программистским слэнгом английского языка, но русскому программисту весьма сложно понять назначение этих имен, а запомнить их могут только люди с феноменальной памятью.

Программирование — одна из наиболее абстрактных областей человеческой деятельности. Пример, приведенный Джесс Либерти и Брэдли Джонс в пользу читабельных имен, работает в обратном направлении. При замене обычных переменных на читабельные теряется общность программы и простая общезначимая формула заменяется на громоздкую, нечитабельную для российского программиста. Можно, конечно, на латинице изобразить русскоязычные названия. Но как этот текст воспримет иностранец? Читабельность программы в наших условиях необходимо улучшать с помощью комментариев, которые являются элементами любого языка программирования.

Вывод 2. Требование конструирования имен соответственно их использованию — необоснованно. Читабельность программы следует поддерживать подробными комментариями, поясняющими запрограммированный алгоритм.

Любой язык программирования включает синтаксис и семантику. Изначально синтаксис многих языков программирования высокого уровня предполагал ввод с перфокарт и имел удобные для этого правила. Синтаксис **Fortran 66** (первый язык программирования высокого уровня) устанавливал строгие правила оформления: 1-я колонка служила для маркировки текста как комментария (символом C), с 1-й по 5-ю располагалась область меток, а с 7-й по 72-ю располагался собственно текст оператора или комментария. Колонки с 73-й по 80-ю могли служить для нумерации карт (чтобы восстановить случайно рассыпавшуюся колоду) или для краткого комментария, транслятором они игнорировались. Если текст оператора не вписывался в отведённое пространство (с 7-й по 72-ю колонку), в 6-ой колонке следующей карты ставился признак продолжения, и затем оператор продолжался на ней. Располагать два или более оператора в одной строке (карте) запрещалось. Когда перфокарты ушли в историю, эти достоинства превратились в серьёзные неудобства. Очевидно, в "перфокарточный период" родилась традиция оформления программ по неким строгим правилам.

Некоторые современные языки программирования также накладывают суровые ограничения на наличие уступов и пробелов. Таким языком является, например, язык функционального программирования **Haskell.** Синтаксис ключевого слова **class** требует через пробел имя класса, тип-параметр и ещё одно ключевое слово **where**. Далее с **отступами** пишутся имена определённых в классе значений.

"*Несколько слов о стиле программирования. Вот эти несколько слов: стиль программирования меня не волнует. Я достаточно краток? Если хотя бы половина времени, израсходованного на правильную расстановку фигурных скобок, тратилась на обдумывание программы или еще лучше — на общение с пользователями, то вся отрасль работала бы*

намного эффективнее. Конечно, единство стиля — вещь хорошая, но я еще не видел книги или руководства по стилю, которые бы стоили даже ***часового собрания группы в начале проекта****. К тому же ни одна книга или руководство по стилю не превратят код неаккуратного программиста в нечто осмысленное. В сущности, стиль часто используется как оправдание недостатка внимания к самой программе. Наконец, я еще не видел, чтобы в спорах о стиле один программист в чем-то убедил другого, поэтому любые дискуссии на эту тему считаю бесполезной тратой времени*." [6] (выделено Рудаковым С.А.).

Вывод 3. При обучении требования оформления исходного кода программы по каким-либо правилам, не следующим логически из синтаксиса языка, — недопустимо. Правила оформления исходного кода программы есть следствие корпоративного соглашения.

Библиографический список

1. *Джесс Либерти, Брэдли Джонс*. Освой самостоятельно C++ за 21 день Изд.: Вильямс ISBN 978-5-8459-0926-8, 0-672-32711-2; 2010 г.
2. http://edu.mmcs.sfedu.ru/file.php/36/cs_coding_std.pdf
3. *М. УЭИТ, С. ПРАТА, Д. МАРТИН.* Язык Си руководство для начинающих, ISBN 5-03-001309-1 /русск./ ISBN 0-672-22090-3 /англ./© 1984 The Waite Group, Inc, © перевод на русский язык: Москва "Мир", 1988
4. http://progopedia.ru/paradigm/
5. http://ru.wikipedia.org/wiki/ Парадигма программирования
6. *Джефф Элджер*. Библиотека программиста C++, 2008, PDF, 320с.

Т.Н. Рудакова

ПРИМЕНЕНИЕ СИСТЕМЫ MATLAB В КУРСЕ "МЕТОДЫ ОПТИМИЗАЦИИ"

rtn@susu.ac.ru

Южно-Уральский государственный университет, Челябинск

The article considers that MatLab is the best informations tehnologies in the educatiom of students to solve ill-posed problems.

Под оптимизацией понимают процесс выбора наилучшего варианта из всех возможных. Практически любые задачи в математике сводятся к задачам оптимизации: безусловная оптимизация нелинейных функций; метод наименьших квадратов; решение нелинейных уравнений; линейное программирование; квадратичное программирование; условная минимизация нелинейных функций; методы минимакса; многокритериальная оптимизация; методы решения некорректных задач. Курс обучения состоит из теоретической части в виде лекций и практических занятий. Для демонстрации примеров во время лекции и для реализации численных алгоритмов на практических занятиях в компьютерном классе используется пакет программ MatLab.

Разработчики системы MATLAB (фирма Math Works, Inc., U.S.A.) учли опыт численного решения и программирования задач вычислительной математики за все время существования вычислительной техники. Поэтому в системе MATLAB по каждой проблеме имеется несколько программ (иногда их более 10), предназначенных для ее решения в зависимости от особенностей данной задачи. Кроме чисто научных задач средствами MATLAB могут быть успешно решены и довольно сложные инженерные проблемы, такие, как поиск спектра частот