Вопрос 3. Оформление программного кода в соответствии с установленными требованиями

1. Спецификация языков программирования

Спецификация *(стандарт, определение)* **языка программирования** — документ, который определяет язык программирования, чтобы пользователи и разработчики могли согласовывать, что означают программы на данном языке. Спецификации обычно являются подробными и формальными. В основном, эти документы используются разработчиками языка, в то время как пользователи обращаются к ним в случае двусмысленности: например, спецификация языка C++ часто цитируется пользователями из-за сложности.

Сопутствующая документация включает справочник по языку программирования, который специально предназначен для пользователей, и логическое обоснование языка программирования, которое объясняет, почему спецификация написана именно так; последние обычно более неформальны, чем спецификации.

Спецификация может включать в себя следующие компоненты:

Стандартизация – приведение языков программирования к международным стандартам, производимое специальными организациями по поводу разработки, обновления, публикации и спецификаций языков программирования.

Стандарт оформления ко́да (станда́рт коди́рования, стиль программи́рования) — набор правил и соглашений, используемых при написании исходного **кода** на некотором языке программирования.

Алфавит – фиксированный для данного языка набор основных символов, допускаемых для составления текста программы на этом языке.

Лексика – фиксированный набор основных слов (команд), определяющих выполняемые компьютером действия.

Синтаксис – система правил, определяющих допустимые конструкции языка программирования из букв алфавита.

Семантика – система правил однозначного толкования отдельных языковых конструкций, позволяющих воспроизвести процесс обработки данных.

Не все основные языки программирования имеют спецификации; языки могут существовать и быть популярными в течение десятилетий без спецификации. Язык может иметь одну или несколько реализаций, поведение которых де-факто является стандартом, но при этом данное поведение не документировано в спецификации.

Perl (Perl 5) — известный пример языка без спецификации, тогда как PHP получил спецификацию только в 2014 году, после использования в течение 20 лет.

Язык может быть реализован, а затем стандартизован, или стандартизован, а после этого реализован, или же два этих процесса могут развиваться вместе, что является в настоящее время обычной практикой. Это связано с тем, что реализация и спецификация обеспечивают проверку друг друга: для написания спецификации требуется точно указать поведение реализации, а реализация проверяет, что спецификация является возможной, целесообразна и последовательна.

Написания спецификации до реализации зачастую стали избегать после инцидента с программой Алгол 68 (1968 г.), из-за неожиданных трудностей с реализации, которая была отложена. Тем не менее, языки всё ещё время от времени используются и приобретают

популярность без формальной спецификации: реализация языка необходима для его использования, в то время как спецификация является желательной, но не необходимой.

Спецификация языков программирования может принимать самые разные формы:

- Явное определение синтаксиса и семантики языка. Хотя синтаксис обычно задается с использованием формальной грамматики, семантические определения могут быть написаны на естественном языке.

Примечательным примером является язык Си, который приобрёл популярность без формальной спецификации, а был описан вместо этого в части книги «Язык программирования Си» (1978 г.), и был формально стандартизован намного позже в ANSI C (1989 г.).

- Описание поведения компилятора для языка (например, языки C++ и Фортран). Синтаксис и семантика языка выводятся из этого описания, которое может быть написано на естественном или формальном языке.
- Эталонная реализация, иногда написанная на указанном языке (например, Пролог). Синтаксис и семантика языка подробно ясны из поведения реализации модели.
- Синтаксис языка программирования обычно описывается с использованием комбинации следующих двух компонентов:

- регулярное выражение, описывающее его лексемы (распознанные группы входной последовательности символов), и
- контекстно-свободная грамматика, которая описывает, как лексемы могут быть скомбинированы для формирования синтаксически правильной программы.

Формулирование строгой семантики большого, сложного, практичного языка программирования является сложной задачей даже для опытных специалистов, и получающаяся в результате спецификация может быть трудной для понимания всех, кроме экспертов. Ниже приведены некоторые из способов описания семантики языка программирования; все языки используют по крайней мере один из этих методов описания, а некоторые языки объединяют более одного:

- естественный язык: описание естественным человеческим языком;
- формальная семантика: математическое описание;
- эталонная реализация: описание посредством компьютерной программы;
- набор тестов: описание при помощи примеров программ и их ожидаемого поведения. Хотя лишь несколько спецификаций языка начинались с этой формы спецификации, семантика набора тестов влияет на эволюцию некоторых спецификаций.

2. Значение стиля оформления кода и его формализация

Умение правильно оформлять программный код – одна из профессиональных способностей разработчика программного обеспечения. Важно не только знать и уметь применять конструкции языка программирования, писать эффективные по времени программы, использовать алгоритмические приёмы и стандартные шаблоны программирования, но и «упаковывать» всё это в корректную и эффективную текстовую оболочку.

Унификация и рациональная структура кода улучшают читабельность и понятность программ, ускоряют тестирование и отладку, упрощают взаимопонимание и взаимодействие в группе разработчиков. Таким образом, следование единым правилам оформления делается

не столько для придания текстам программ красивого внешнего вида, сколько для повышения производительности разработки и качества программного продукта.

Для разных языков программирования существует разная история формирования и принятия правил оформления исходного кода. В большинстве случаев единых общемировых требований и рекомендаций не существует, но складываются определенные традиции и неформальные соглашения в сообществах разработчиков, авторами книг и статей используются общепринятые правила кодирования.

Для каждого языка программирования в той или иной форме существуют соглашения и правила по написанию кода: в некоторых случаях это формальные документы (как для языка Java), в других – сложившиеся полуофициальные традиции профессиональных сообществ разработчиков или корпоративные регламенты.

Например, *для языка Java в 1995 году было принято и опубликовано соглашение по оформлению кода, следование которому является де-факто хорошим тоном для всего сообщества java-программистов. Для некоторых языков программирования источником стандартов оформления является компания-разработчик языка, например Microsoft выпущены рекомендации для языка C#.*

Язык программирования C (иначе - Си) был разработан в 1960-1970-х годах сотрудником компании Bell Labs Деннисом Ритчи. Язык C++ создавался в начале 1980-х на основе C с использованием объектноориентированной парадигмы, его автором является Бьёрн Страуструп, разработчик той же компании. Не существует принятых общемировых стандартов или рекомендаций по оформлению кода на C и C++. Есть корпоративные, проектные и университетские соглашения по стилю кодирования на этих языках, в частности можно назвать следующие документы:

1. Русско-венгерская нотация.

2. руководство от факультета компьютерных наук Стенфордского университета по стилю оформления кода на C++ в рамках учебного курса CS 106B.

3. руководство от компании Google по стилю программирования на C++.

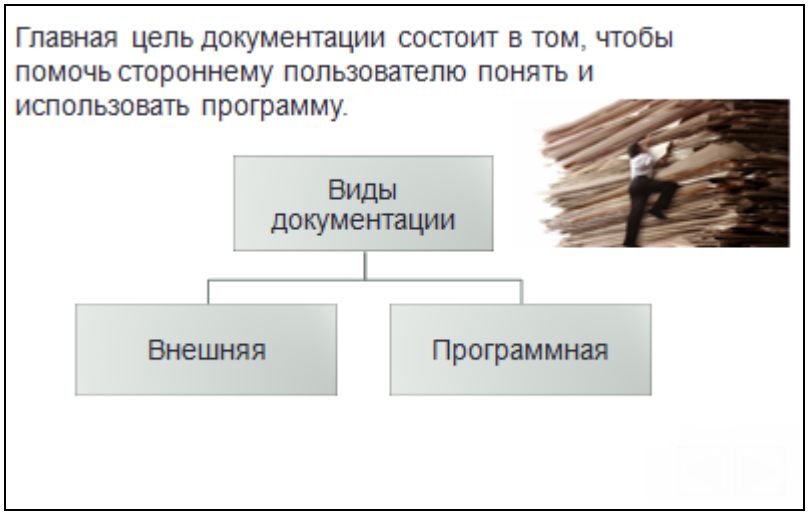
Требования и рекомендации по оформлению программного кода на C/C++ основаны как на сложившейся в компьютерной литературе и прикладном программировании практике, так и на указанных выше документах и соглашениях. Так как синтаксис языка C++ основан на синтаксисе языка C, то в части требований к оформлению программ эти два языка практически не отличаются.

Соблюдение единых правил оформления программного кода значительно облегчает читаемость кода и позволяет обеспечивать высокую эффективность технической поддержки информационных систем и процессов реализации доработок в информационных системах.

Правила, стандарты и соглашения по оформлению программного кода на языке программирования (coding standard, coding convention, programming style, code style guide) необходимо осваивать одновременно с самим языком. Единый стиль написания текстов программ облегчает понимание, упрощает взаимодействие разработчиков программного обеспечения, значительно ускоряет процессы тестирования и отладки.

При начальном обучении программированию важно изучение правил оформления кода на самых ранних этапах для последующего формирования и закрепления профессиональных навыков по строгому следованию требованиям написания программ.

3. Виды программной документации



Внешняя документация представляет собой сведения о программе, не содержащиеся в самой программе. В зависимости от размеров и сложности программы может принимать различные формы:

- схемы или словесные описания алгоритмов;
- инструкции для пользователей;
- образцы входных и выходных данных;
- полное описание процесса построения программы;
- ссылки на источники информации и др.

Программная документация реализуется с помощью комментариев (в начале программы – в виде заголовка и вводных комментариев; поясняющих — внутри текста программы), а также рационального выбора имён, применения стандартных методов структурирования программ.

Компании - производители программного обеспечения часто используют собственные стандарты разработки программного обеспечения, где прописаны внутрикорпоративные требования по оформлению кода и комментированию программных кодов информационных систем. Такие внутренние документы определяют:

- основные требования к оформлению программных кодов при разработке программных кодов для любых информационных систем организации;

- принципы построения программного кода в части их единого структурирования по определенным правилам;

- формат содержательных пояснений (комментариев) к программным кодам.

Такой стандарт является обязательным для любых разработок новых информационных систем, а также любых доработок существующих информационных продуктов в рамках организации. Отклонение от таких стандартов возможно только по предварительному письменному разрешению руководителя, курирующего направление разработки. В некоторых случаях такие требования формулируются для отдельных проектов.

Необходимо отличать принимаемые международные стандарты языков программирования (ANSI/ISO, ISO/IEC) от совокупности правил, соглашений и стандартов оформления программного кода. В первом случае стандарт является основой для реализации языка как инструментального средства разработки, то есть создания транслятора и его инфраструктуры. Во втором – это по сути соглашения и правила рекомендательного характера для разработчиков, следование которым не является обязательным с точки зрения синтаксиса или семантики языка, но значительно повышает качество и удобство разработки программных продуктов.

4. Правила оформления программной документации в России

Программная документация является неотъемлемым компонентом программного продукта и должна оформляться в соответствии с Единой системой программной документации (ЕСПД — ГОСТ серии 19).

Программная документация, кроме формальных документов (спецификация, ведомость держателей подлинников, формуляр и др.), включает:

- техническое задание (назначение, область применения программы, требования, предъявляемые к программе);
- текст программы (запись программы с необходимыми комментариями);
- описание программы (сведения о логической структуре и функционировании программы);
- пояснительная записка (схема алгоритма, общее описание алгоритма и/или функционирования программы, обоснование принятых решений);
- эксплуатационные документы.

Разделы технического задания:

- введение (наименование, краткая характеристика области применения программы);
- основания для разработки (документы, на основании которых ведётся разработка, организация, утвердившая документы, дата утверждения, наименование и обозначение темы разработки);
- назначение разработки (функциональное и эксплуатационное назначение программы);
- требования к программе и программной документации;
- технико-экономические показатели;
- стадии и этапы разработки;
- порядок контроля и приёмки.

Наиболее существенной частью технического задания является раздел «требования. » В этом разделе приводятся:

- требования к функциональным характеристикам (состав выполняемых функций, организация входных и выходных данных, временные характеристики);
- требования к надёжности (обеспечение устойчивого функционирования, контроль входной и выходной информации, время восстановления после отказа);
- требования к информационной и программной совместимости (требования к информационным структурам на входе и выходе, методам решения, исходным кодам, языкам программирования и программным средствам; требования к защите информации);
- требования к составу и параметрам технических средств;
- требования к программной документации.

Данный раздел может содержать требования к маркировке, упаковке, транспортировке и хранению, а также условия эксплуатации.

Кроме явно описанных в техническом задании требований, следует придерживаться общепринятых правил разработки программ с учётом выбранной парадигмы программирования:

1. Программа не должна содержать избыточные элементы (все элементы программы адекватны поставленной задаче: нет циклов, массивов и т. п. элементов, без которых можно обойтись).
2. Алгоритм должен быть структурирован: для функционального стиля программирования — адекватное разбиение на функции (процедуры), для объектно-ориентированного — адекватная иерархия классов. Каждая функция (метод класса) должна реализовывать ровно одно действие.
3. У функций (методов классов) должны быть параметры. Следует избегать использования в функциях глобальных переменных.
4. Программа должна аккуратно использовать память: работать с динамическими массивами, в ней не должно быть неиспользуемых блоков памяти, лишних переменных.
5. Должны проверятся диапазоны вводимых пользователем значений и параметров, передаваемых между модулями программы.
6. При использовании в программе каких-либо готовых компонент (библиотечных функций, классов) если функция или метод класса может завершиться неудачей, необходимо обязательно проверять это, не полагаясь на незначительность вероятности такого события.
7. Программа должна быть конфигурируема (важные параметры программы следует выделить в единый блок).

Текст программы представляет собой символическую запись на исходном или промежуточном языке или символическое представление машинных кодов. Текст программы оформляется моноширинным шрифтом (Courier, Lucida Console и т. п.) в соответствии с общепринятыми нормами оформления.

Документ «Описание программы» содержит:

- общие сведения (обозначение наименование программы, программное обеспечение, необходимое для функционирования программы, языки программирования, на которых написана программа);
- функциональное назначение (классы решаемых задач, сведения о функциональных ограничениях на применение);
- описание логической структуры (алгоритм программы, используемые методы, структура программы с описанием составных частей и связи между ними);
- используемые технические средства (типы ЭВМ и устройств, которые используются при работе программы);
- вызов и загрузка (способ вызова программы с соответствующего носителя данных);
- входные данные (характер, организация и предварительная подготовка входных данных, а также их формат, описание и способ кодирования);
- выходные данные (характер и организация выходных данных, а также их формат, описание и способ кодирования).

Описание логической структуры программы следует сопровождать блок-схемой программы.

Документ «Описание программы» может содержать также схемы данных, схемы взаимодействия программ, схемы ресурсов системы и проч., оформленные в соответствии с ГОСТ 19.701-90.

Пояснительная записка составляется на стадии эскизного или технического проектов программы. Как правило, на стадии рабочего проекта не используется.

К эксплуатационным документам относят:

- описание применения (сведения о назначении программы, области применения, применяемых методах, классе решаемых задач, ограничениях для применения, минимальной конфигурации технических средств);
- руководство системного программиста (сведения для проверки, обеспечения функционирования и настройки программы на условия конкретного применения);
- руководство программиста (сведения для эксплуатации программы);
- руководство оператора (сведения для обеспечения общения оператора с вычислительной системой в процессе выполнения программы);
- описание языка (описание синтаксиса и семантики языка);
- руководство по техническому обслуживанию (сведения для применения тестовых и диагностических программ при обслуживании технических средств)

Основная часть программной документации составляется на стадии рабочего проекта. Необходимость того или иного документа определяется на этапе составления технического задания. Допускается объединять отдельные виды документов.

Литература:

1. Спецификация языков программирования: статья в Википедии / https://ru.wikipedia.org/wiki/%D0%A1%D0%BF%D0%B5%D1%86%D0%B8%D1%84%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D1%8F_%D1%8F%D0%B7%D1%8B%D0%BA%D0%BE%D0%B2_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F
2. Анисимов А.Е. Требования и рекомендации по оформлению программного кода на языках C и C++ / А.Е. Анисимов.– Ижевск: Издательский центр «Удмуртский университет», 2020. – 48 с. (стр. 1-7) / http://elibrary.udsu.ru/xmlui/bitstream/handle/123456789/19392/325%D0%BB%D0%B1_1000984252_11.08.2020.pdf?sequence=1
3. Стандарт разработки программного обеспечения и комментирования программных кодов информационных систем НИУ ВШЭ / https://www.hse.ru/data/2015/04/30/1098219251/%D0%A1%D1%82%D0%B0%D0%BD%D0%B4%D0%B0%D1%80%D1%82%20%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B8%20%D0%B8%20%D0%BA%D0%BE%D0%BC%D0%BC%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F%20%D0%9F%D0%9E.pdf
4. Гост программный код. Правила оформления программной документации / https://tractorillo.ru/gost-programmnyi-kod-pravila-oformleniya-programmnoi-dokumentacii-vidy/