

СОВМЕСТНОЕ ИСПОЛЬЗОВАНИЕ ВИЗУАЛЬНЫХ НОТАЦИЙ UML И Р-СХЕМ В ПРОЦЕССЕ ОБУЧЕНИЯ ОБЪЕКТНО- ОРИЕНТИРОВАННЫМ МЕТОДАМ ПРОГРАММИРОВАНИЯ

Дробушевич Л.Ф., Конах В.В.

Инновационные процессы, происходящие сегодня в образовании, обуславливают необходимость подготовки специалистов, готовых работать в изменившихся условиях, способных осваивать современные методы и профессиональные инструменты,

Одной из основных целей университетского курса «Технологии программирования» является обучение студентов непрерывному процессу приближения к коду программной системы [1]. В общем виде процесс начинается с понимания предметной области, выявления задач в виде прецедентов, описания работы функций прецедентов и заканчивается в конечном счете реализацией проекта.

При таком понимании процесса разработки программных систем (ПС) особая роль отводится использованию формализованных методов анализа и проектирования. Имеется ряд программных инструментов, которые включают возможность моделирования с использованием языка UML (Visio от Microsoft, Rational Software Architect от IBM, Enterprise Architect от Sparx System и другие).

С помощью моделей [1,2] по-новому решаются проблемы документирования, тестирования, отладки, организации коллективной работы специалистов. Модели, *во-первых*, важны, ибо, в большей степени состоят из рисунков, которые могут передать намного больше информации, чем большое количество текста или кода. *Во-вторых*, модели ценны, потому что создание их и поддержка обходится дешевле. *В-третьих*, вносить изменения в модели проще и легче, чем в код. А дешевизна, быстрота и гибкость – это именно то, что необходимо при решении практических задач.

Главное преимущество использования визуальных моделей в учебном процессе заключается в том, что с их помощью легче объяснять алгоритмы, проще их записывать, а значит, и анализировать, легче понимать и инспектировать логическую структуру ПО и вносить соответствующие изменения.

Предлагается в процессе обучения объектно-ориентированной методологии совместно использовать две графические нотации – UML и графические структуры Р-схем [3]. Р-схемы являются

нагруженными по дугам графами, причем нагрузки дуг могут записываться в любом виде, в том числе и на языке программирования. В этом смысле Р-схемы можно рассматривать в качестве фундамента записи любых алгоритмов, процессов, программ. В результате формируется другой стиль проектирования и программирования, основанный на более естественном, рациональном и защищенном от многих ошибок мышлении специалистов, с лучшим использованием мощного зрительного аппарата человека.

Эффективное совместное использование UML и Р-схем, как показала практика обучения объектно-ориентированным методам разработки ПО, методологически возможно в *двух аспектах: вместо и дополнительно*:

1. Р-схемы *вместо* диаграмм Деятельности (Activity diagram).

Одной из первых диаграмм UML, с которой приходится иметь дело в процессе анализа предметной области является диаграмма Деятельности. Диаграмма Деятельности в нотации UML очень близка к блок-схеме, даже используемые в ней обозначения аналогичны. В отличие же от блок-схем диаграммы Деятельности могут моделировать параллельно происходящие процессы, что немаловажно для исключения избыточных процессов при моделировании. Но создавать и редактировать диаграммы Деятельности, как прочем, и “старые добрые” блок-схемы не совсем просто в традиционных инструментах, поддерживающих UML. По этой причине студенты, да и разработчики программных проектов используют их в процессе моделирования довольно редко. Поэтому целесообразнее *вместо* диаграмм Деятельностей использовать Р-схемы, которые проще рисовать, редактировать и понимать.

2. Р-схемы *дополнительно* к диаграммам UML на этапе детального проектирования для спецификации алгоритмов функций системы и методов классов.

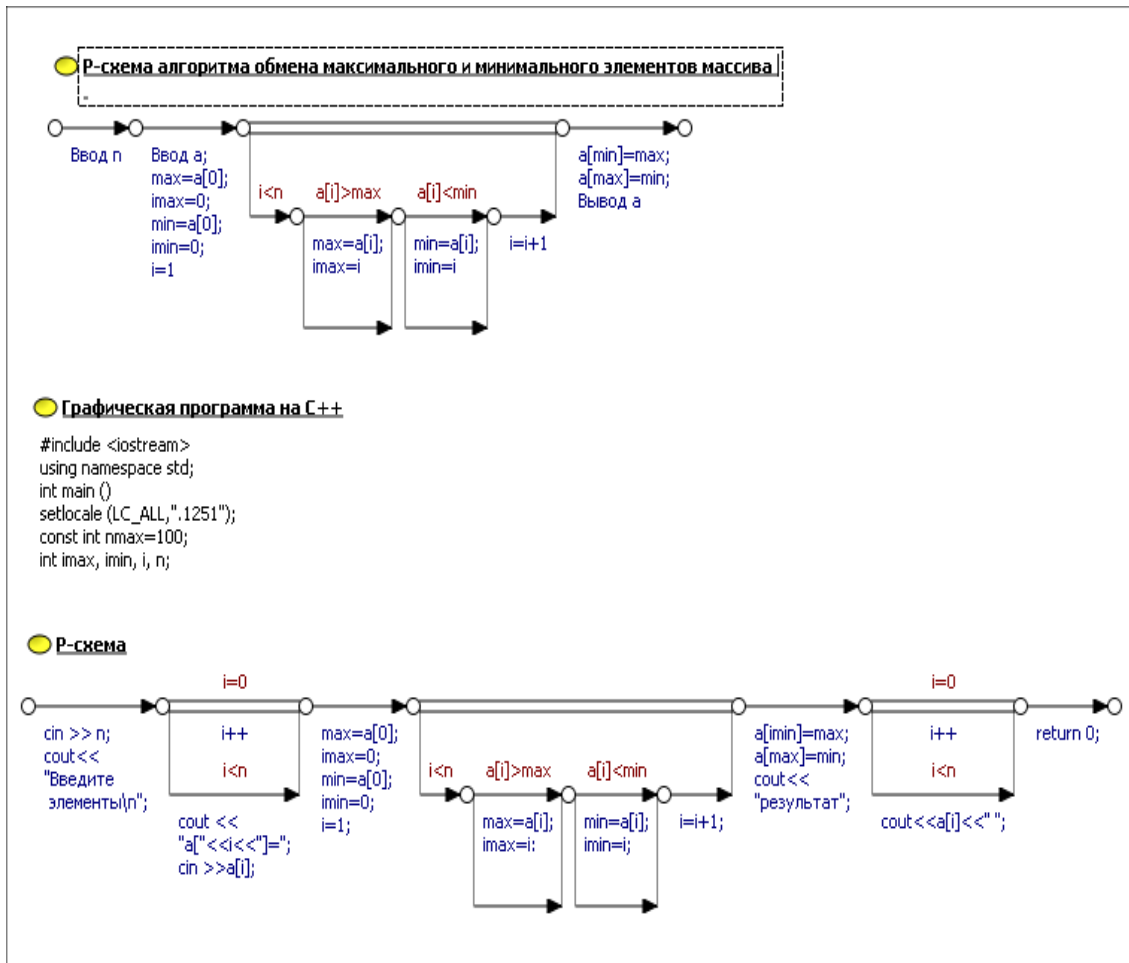
В настоящее время существует несколько инструментальных систем, поддерживающих процесс визуального моделирования с помощью нотации UML, позволяющих получать по элементам модели каркасы исходных кодов на любых объектно-ориентированных языках программирования. Однако ни один из них не поддерживает визуализацию на этапе реализации функций или методов классов разрабатываемой системы. Поэтому предлагается использовать Р-схемы в качестве визуального средства построения кода, то есть логику реализации функций представлять в виде Р-схем. Потом

встроенный в среду проектирования (или моделирования) графический редактор автоматически переведет Р-схемы в традиционные исходные коды на языке программирования. Таким образом Р-схемы могут оказаться очень полезными, если не для всего приложения, то для отдельных сложных и ответственных его частей, что в итоге приведет к повышению качества и надежности проекта в целом.

Использование сквозной визуализации в процессе разработки программных проектов позволит повысить качество и надежность программ, т.к. на этапе кодирования программисту не придется заниматься мелкими деталями. При этом основной *механизм абстрагирования* объектно-ориентированного подхода будет активно использоваться разработчиком на всех этапах разработки программного проекта: от эскизного до детального проекта и далее – кода реализации. Такой подход позволяет работать на более высоком уровне абстракции и обеспечивает *простой и бесшовный* переход от проекта к коду.

Одним из преимуществ такого подхода является то, что разработчик может *проинспектировать алгоритм* отдельно от программы без углубления в сложности и детали языка программирования. Благодаря этому некоторые дефекты могут быть обнаружены до того, как они перерастут в дефекты программного кода. Чем важнее метод, тем важнее эта детальность. Методы со сложными ветвлениями и циклами являются первыми кандидатами для построения Р-схем. Инспектирование до программирования – основная идея использования Р-схем на данном этапе разработки.

Второе преимущество данного подхода состоит в том, что можно обеспечить *функцию поддержки кода и модели* алгоритма в актуальном состоянии. Причем такая функция может работать в реальном масштабе времени. Создав Р-схему алгоритма возможно тут же сгенерировать по ней исходный код со всеми деталями и тонкостями языка программирования. И, наоборот, имея исходный код, можно получить по нему Р-схему алгоритма. Причем, если реализовать (в будущем) в редакторе режим синхронизации, инструмент будет делать это автоматически. Визуальная схема алгоритма реализованного метода в дальнейшем может оказать разработчику большую помощь в процессе тестирования и отладки программ. В подтверждение сказанного рассмотрим простейший пример.



Логика алгоритма и графической программы абсолютно совпадают. Таким образом, хотя Р-схемы не являются составной частью UML, они могут эффективно *использоваться совместно*, чтобы помочь делать качественные проекты. Благодаря этому некоторые дефекты программного проекта могут быть обнаружены до того, как они перерастут в дефекты программного кода. Чем сложнее проект, чем больше логических ветвлений в нем и в выделенных функциях разрабатываемой системы, тем эффективнее использование Р-схем.

ЛИТЕРАТУРА

1. Буч, Г. Язык UML: Руководство пользователя / Г. Буч, Дж. Рамбо, А. Якобсон; пер. с англ. М., 2000. 432с.
2. Дробушевич, Л. Визуальное моделирование программных систем с помощью Enterprise Architect 7.0 и UML / Л. Дробушевич, А. Исаченко. Минск: БГУ, 2009. 160 с..
3. Вельбицкий, И. В. Визуальная технология программирования нового поколения для широкого применения на базе стандарта ISO/IEC 8631 / И. В. Вельбицкий // Межд. конф. MEDIAS - 2010, Кипр, 2010.