

Программируемая анимация широкое применение в сети получили два языка, с помощью которых программируются движения анимируемых объектов: Java-Script (браузерный язык) и Action-Script (язык работы с приложениями Flash). Преимущество программируемой анимации – в уменьшении размера исходного файла, недостаток – нагрузка на процессор клиента.

Для создания анимированных изображений существует множество различных программ, например: Adobe Photoshop, Adobe Flash, GIMP, CoffeeCup, Synfig и другие.

Компьютерная анимация может храниться в универсальных графических файлах (например, в формате GIF) в виде набора независимых изображений, либо в специализированных файлах соответствующих пакетов анимации (3ds Max, Blender, Maya и т. п.) в виде текстур и отдельных элементов, либо в форматах, предназначенных для просмотра (FLIC) и применения в играх (Bink). Также, анимация может сохраняться в форматах, предназначенных для хранения видео (например, MPEG-4).

Компьютерная анимация на сегодня получила широкое применение в компьютерных играх, мультимедийных приложениях, а также для «оживления» отдельных элементов оформления, например, веб-страниц и рекламы. С середины 1980 годов компьютерная анимация используется для создания спецэффектов в кинематографе.

УДК.621.762.4

Оскирко А. С.

## **ЗАКОНЫ ПРОГРАММИРОВАНИЯ**

*Научный руководитель Дробыш А. А.*

Наличие законов – неотъемлемая часть нашей жизни. Тоже самое можно сказать и про науку, и информатику в частности.

В информатике неотъемлемую роль играет программирование, и у него также есть свои законы и порядки. Целью моей работы стал анализ разнообразных информационных ресурсов, как книжных, так и электронных, чтобы узнать, какие существуют законы в программировании.

Приведем один из законов программирования: «Закон необходимости ошибок. Программист может обнаружить ошибку только в чужой программе». С одной стороны, эта фраза похожа на юморную шутку, но существует такое высказывание: «В каждой шутке есть доля правды». И в данном законе также присутствует часть правды, так как действительно можно обратить внимание на то, что практически каждый сторонний программист может обнаружить ошибку, о которой вы не подозревали, или которая не давала вам спокойно прокоопмилировать код. Так происходит, потому что у стороннего программиста более «свежий» взгляд на вашу работу, и он обращает внимание на те участки кода, где, по его мнению, может быть ошибка.

Следующий закон программирования можно трактовать так: «Любая программа обходится дороже и требует больших затрат времени, чем предполагалось». В суть этого закона заложено то, что программист, насколько бы тщательно он не спланировал свои действия, будет затрачено больше ресурсов и времени. А если вы сделали программу раньше положенного срока, тогда следует ее еще раз проверить на наличие ошибок и недочетов.

Закон возникновения ошибок: «Если отладка – процесс удаления ошибок, то программирование должно быть процессом их внесения». Сущность этого закона состоит в том, что процесс отладки программы позволяет ликвидировать найденные ошибки, и по логике можно сказать, что ошибки возникают от программиста, вернее от его работы в виде написания кода.

Фредерик Брукс-младший придумал высказывание, которое звучит так: «Выдавать глобальные идеи – это удовольствие; искать сволочные маленькие ошибки – вот настоящая работа». Это высказывание заставляет задуматься о том, что глобальная идея может прийти в голову спонтанно, а нахождение ошибок в коде – это кропотливый труд.

Итак, проанализировав разные источники, может прийти к выводу, что законы программирования существуют и будут использоваться на протяжении многих лет.

УДК 621

Романенко А. С.

## **БИТОВЫЕ ПОЛЯ И ОПЕРАЦИИ НАД НИМИ**

*БНТУ, Минск*

*Научный руководитель Дробыш А. А*

Битовое поле – некоторое количество бит, расположенных последовательно в памяти, значение которых процессор не способен прочесть из-за особенностей аппаратной реализации. Битовые поля могут оказаться полезными во многих ситуациях:

- если память ограничена, в одном байте можно хранить несколько логических переменных, принимающих значение true и false;
- когда в одном байте нужно хранить информацию о состоянии некоторых устройств, закодированную несколькими битами;
- если шифровальным процедурам требуется доступ к отдельным битам.

Порядок размещения битовых полей в памяти в значительной степени зависит от компьютера и аппаратного обеспечения. Поля могут описываться только в качестве элементов