Таблица 2

SWOT-анализ процесса «ВА СМК в ОрГМУ»

(S) Сильные стороны	(W) Слабые стороны
1. Проведение аудита квалифицированными специалистами 2. Минимальное вложение денежных средств на проведение ВА 3.. Независимость проведения ВА 4. Отсутствие настороженного отношения сотрудников проверяемых подразделений к аудиторам	1. Отсутствие единой шкалы для оценивания результатов проведения ВА 2. Выявление несоответствий в деятельности подразделений аудиторами воспринимаются болезненнее 3.Человеческий фактор 4.Отсутствие критериев оценки кандидатов в аудиторы
5. Знание аудиторами всех особенностей организации 6. Разработка рекомендаций по дальнейшему улучшению деятельности СМК 7. Возможность более полного анализа критериев аудита (по статистике внутренний аудитор обнаруживает несоответствия от 10 до 100 раз больше, чем внешний аудитор).	5. Незаинтересованность высшего Руководства 6. Отсутствие критериев оценки кандидатов в аудиторы 7. Отсутствие единой шкалы оценивания результатов проведения ВА
(O) Возможности	(T) Угрозы
1. Подготовка внутренней документации к внешнему аудиту 2. Анализ и устранение причин выявленных несоответствий 3. Подтверждение соответствия деятельности и ее результатов в СМК установленным требованиям 4. Подтверждение выполнения корректирующих действий по результатам предыдущих аудитов 5. Определение путей дальнейшего совершенствования	1. Недостаточная компетентность аудиторов 2. Не результативность ВА 3. Срыв внутреннего аудита 4. Ошибки в передаче информации об аудите 5. Нарушение плана работ проверяемого подразделения 6. Создание стрессовой ситуации для работников проверяемых подразделений

Как правило, применение SWOT-анализа позволяет систематизировать имеющуюся информацию и на основе этого принимать результативные решения, касающиеся процесса «ВА СМК». Кроме этого на основе преимуществ использования SWOT-анализа, представленных, например, в разработанном на кафедре МСиС пособии [5], нами определены те стороны процесса «ВА СМК», которые необходимо совершенствовать в первую очередь и те, которые представляют собой стратегическую перспективу. Особое внимание мы считаем необходимо уделять человеческому фактору, независимости проведения аудита и высокой вероятности не выявления областей для совершенствования аудируемого процесса.

Таким образом, элементы ТТО в обеспечении качества процесса «ВА СМК в ОрГМУ» предлагается рассматривать как необходимые составляющие факторы данного процесса. Анализ, систематизацию элементов ТТО необходимо проводить с учетом особенностей этапов жизненного цикла процесса. При этом особое внимание следует уделять разработке и актуализации нормативных документов.

Работа выполнена под руководством заведующего кафедрой метрологии, стандартизации и сертификации – академика РАЕ, д-р техн. наук, доцента Третьяк Л.Н.

Список литературы

1. Косых Д.А. Внутренний аудит систем менеджмента качества [Электронный ресурс]: электронный курс в системе Moodle / Д.А. Косых, Л.Н. Третьяк, В.А. Гарельский; М-во науки и высш. образования Рос. Федерации, Федер. гос. бюджет. образоват. учреждение высш. образования «Оренбург. гос. ун-т». Оренбург: ОГУ, 2020. 7 с.

2. Чикачек Е.В. К вопросу о необходимости внедрения системы менеджмента качества в медицинские образовательные учреждения // Международный студенческий научный вестник. 2020. № 2. С. 23-28.

3. ГОСТ Р 15.000-2016. Система разработки и постановки продукции на производство. Основные положения. М.: Стандартинформ, 2016. 19 с.

4. Третьяк Л.Н. Роль технико-технологических элементов в обеспечении качества пива с заданными свойствами [Электронный ресурс] / Л.Н. Третьяк, В.В. Гагауз // Международный студенческий научный вестник, 2020. № 2. С. 71. 14 с.

5. Пыхтин А.В. Статистические инструменты контроля качества [Текст]: практикум / А.В. Пыхтин, В.А. Лукоянов; М-во образования и науки Рос. Федерации, Федер. гос. бюджет. образоват. учреждение высш. проф. образования «Оренбург. гос. ун-т». Оренбург: Университет, 2013. 104 с.

СОЗДАНИЕ СОБСТВЕННЫХ БИБЛИОТЕК В SCILAB

Ярошко Я.С., Чижикова Е.С.

Тюменский индустриальный университет (филиал в г. Тобольске), e-mail: nerdcapricorn@mail.ru

Библиотека представляет собой набор функций, написанных на языке Scilab и хранящихся в отдельных файлах, и является самой простой

организацией программируемых функций в среде. Платформа Scilab предоставляет возможности разработки и создания собственных пакетов расширений по мимо уже предоставленных разработчиками. Во многих практических ситуациях при многократном повторении однотипных или схожих алгоритмов действий пользователь сможет воспользоваться пакетом заранее заготовленных функций и команд с целью упрощения и оптимизации рабочего процесса.

Scilab (англ.) – с одной стороны, это математический пакет численных расчетов, а с другой – платформа для разработки высокотехнологичных приложений, где имеются все атрибуты среды разработки: текстовый редактор, компилятор и интерпретатор, язык программирования, средства отладки [3].

Пакет расширений может состоять из функций, написанных как на встроенном языке программирования, так и на сторонних языках. Пакет, поставляемый конечному пользователю, может состоять из текстов функций вместе со скриптами сборки и загрузки, или включать только уже скомпилированные файлы и скрипт их загрузки в систему. В данной статье мы рассмотрим создание пакета, содержащего функции на встроенном языке программирования Scilab [2].

Библиотека (англ. library) – это организованный набор функций, написанных на языке Scilab (т.е. являющихся макросами), хранящийся на нескольких файлах [1]. Как правило библиотеки являются составляющими элементами при построении модулей, которые могут также содержать файлы справки и файлы с примитивами.

Цель исследования: изучить возможности платформы Scilab и создать собственную пользовательскую библиотеку.

Для решения поставленных задач были выбраны следующие **методы исследования:** систематизация специальной литературы и мировых информационных ресурсов по проблеме исследования.

В библиотеку объединяются наборы функций, связанные между собой некоторой идеей. К преимуществам библиотеки можно отнести следующее:

– Библиотеки загружаются и выгружаются целиком, что защищает все функции входящие в нее.

– Библиотеки легко обслуживать, так как каждая законченная функция хранится в своем файле.

– Разрабатывать библиотеку может группа разработчиков, так как функции соединяются воедино только на этапе сборки.

– С точки зрения производительности, функции скомпилированной библиотеки работают быстрее.

Сам этап сборки библиотеки состоит главным образом из вызова функции genlib(). Сборка должна проводиться в отдельном каталоге, в котором необходимо разместить sci-файлы с функциями. Полный прототип функции genlib() имеет вид:

Первый прототип:

genlib(lib_name [[,dir_name, [Force [,verb [,Names]]]])

Второй прототип:

genlib(lib_name [,path=dir_name] [,verbose=verb] [,force=Force] [,names=Names])

Обозначения:

lib_name – строка; имя библиотеки

dir_name – строка; полный путь до каталога в используемой файловой системе, в котором хранятся sci-файлы функций

Force – логическая константа; в случае %t сценарий перекомпилирует все функции в sci-файлах. По умолчанию установлена %f. Эта возможность используется, когда скомпилированная версия функции не соответствует коду sci-файла, например, когда вы его немного модернизировали.

verb – логическая константа; в случае %t процесс сборки будет сопровождать вывод информации о состоянии в командное окно.

Names – вектор из строк, которые являются именами функций в библиотеки. По умолчанию будут просмотрен весь каталог dir_name, и имена sci-файлов будут приняты за имена функций.

Все файлы, входящие в пакет, должны находиться в одном каталоге, имя которого должно совпадать с названием пакета. Внутри базового каталога пакета находится восемь подкаталогов, каждый из которых имеет свое предназначение. После сборки библиотеки в каталоге появятся новые файлы:

– bin-файлы, которые являются скомпилированными версиями функций в sci-файлах. Из этих файлов и подгружаются функции, поэтому sci-файлы, которые будут находиться рядом с ними, будут фактически уже не нужны. Тем не менее, не следует удалять sci-файлы, потому что они могут вам понадобится в будущем для отладки или для оптимизации, или просто для того, чтобы освежить в памяти применяемые алгоритмы.

– файл names – служебный файл, который содержит имена функций библиотеки. Используется во время подгрузки специальной функцией.

– файл lib – служебный файл, который используется сценарием подгрузки и фактически является головой библиотеки.

Перед непосредственной сборкой следует придерживаться следующих правил:

1. Каждая законченная функция должна хранится в своем sci-файле, и имя этой функции должно в точности совпадать с именем файла. Например, если функция имеет имя function123, то и файл должен быть function123.sci.

2. Внутри sci-файла может быть несколько функций, но только та функция, которая находится в голове файла будет доступна для вызова в среде. Другие функции считаются вспомогательными и могут взаимодействовать только с головной в пределах данного файла.

3. Наличие расширения sci желательно всегда, так как это облегчает работу функции genlib().

Для загрузки готовой библиотеки в среду используется функция lib(). Ее вызов:

namelib = lib('lib_dir')

lib_dir – каталог, в котором хранится собранная библиотека

Функция lib() ищет файлы lib и names и с помощью них подгружает библиотеку. В случае успеха функция вернет имя загруженной библиотеки. Для выгрузки используется функция clear.

Каталог со sci-файлами тоже можно назвать библиотекой. Выгрузить такую библиотеку можно с помощью функции getd(), которая работает примерно как и lib(), только проходится непосредственно по sci-файлам. Однако, такой подход имеет ряд недостатков:

1. не скомпилированные функции работают медленнее;

2. выгрузить функции все сразу уже невозможно, необходимо применять clear для каждой функции в отдельности.

Всегда следует стремиться объединять функции в библиотеки из-за удобства такой организации. Файлы sci рекомендуется применять на начальных этапах, когда реализуемый вами алгоритм еще не завершен.

Перед использованием функций пакета необходимо создать скрипт их сборки, который будет вызываться из основного скрипта всего пакета. Ранее в каталоге с описаниями функций необходимо было также создавать скрипт загрузки, теперь эти функции перешли к файлу с расширением .start, располагающемуся в подкаталоге etc.

Скрипт сборки функций, написанных на встроенном языке, располагается в каталоге macros и, согласно внутренним соглашениям, носит имя buildmacros.sce. Он пишется автором пакета самостоятельно. В Scilab5 появились специальные команды, упрощающие выполнение данной операции. При использовании Scilab на ОС Windows, во время установки необходимо дополнительно выбрать пункты: Parameters Toolbox, Development Tools и A toolbox skeleton – или просто выбрать полную установку [1].

```
C:/samplelib/fuctionl.sci:
function y = functionl(x)
y = 1 * functionl_support(x)
endfunction
function y = functionl_support(x)
y = 3*x
endfunction
2) C:/samplelib/fuction2.sci:
function y = function2(x)
y = 2*x
endfunction
```

Рис. 1. Алгоритм написания собственной простой библиотеки из двух файлов

Следующий пример демонстрирует, как создать бинарную версию функций, воспользовавшись инструкцией genlib. Первый аргумент функции genlib представляет название будущей библиотеки, а второй указывает каталог, где размещены файлы функций. Заметим, что в данном случае только функции functionl и function2 являются общедоступными, а функция functionl_support может использоваться только внутри библиотеки, но не вне ее.

```
genlib("mylibrary", "C:/samplelib")
mylibrary
mylibrary =
Functions files location : C:\samplelib\.
functionl function2
```

Рис. 2. Бинарная версия функций

Функция genlib генерирует и помещает в каталог «C:/samplelib» следующие файлы:

– functionl.bin: бинарная версия файла functionl.sci,

– function2.bin: бинарная версия файла function2.sci,

– lib: бинарная версия библиотеки

– names: текстовый файл, содержащий имена всех функций в библиотеке.

Скомпилированные файлы *.bin и файл lib являются кроссплатформенным в том смысле, что могут без изменений использоваться версиями Scilab для Windows, Linux или Mac OS.

Вместе с тем, на практике нет необходимости каждый раз генерировать библиотеку заново. Готовую библиотеку можно загрузить посредством команды lib, единственный аргумент которой указывает местоположение загружаемой библиотеки в файловой системе. Следующий фрагмент иллюстрирует загрузку ранее созданной библиотеки.

```
mylibrary = lib("C:\samplelib\")
ans =
Functions files location : C:\samplelib\.
functionl function2
```

Рис. 3. Загрузка ранее созданной библиотеки

При большом числе загружаемых библиотек, удобно поместить инструкции lib в стартовом скрипте Scilab, который автоматически исполняется при загрузке пакета. В этом случае все указанные библиотеки будут доступны сразу же после старта Scilab. Файл стартового скрипта размещается в основном каталоге Scilab.

Заключение. В ходе данного исследования на базе платформы Scilab, представляющей собой, пакет прикладных математических программ с открытым окружением для инженерных и научных расчётов, был рассмотрен пример создания собственных независимых библиотек, облегчающих работу пользователя. Во многих практических ситуациях при многократном повторении однотипных или схожих алгоритмов действий пользователь сможет воспользоваться пакетом заранее заготовленных функций и команд с целью упрощения и оптимизации рабочего процесса.

Список литературы

1. Андриевский А.Б., Андриевский Б.Р., Капитонов А.А., Фрадков А.Л. Решение инженерных задач в среде Scilab. Учебное пособие. СПб.: НИУ ИТМО, 2013. 97 с.

2. Официальный сайт разработчика Scilab [Электронный ресурс]. Режим доступа: URL: https://www.scilab.org/ (дата обращения: 16.11.2020).

3. Библиотеки функций [Электронный ресурс]. Режим доступа: URL: http://vse-o-scilab.narod.ru/index/0-52 (дата обращения: 17.11.2020).

4. Scilab/Программирование [Электронный ресурс]. Режим доступа: URL: https://ru.wikibooks.org/wiki/Scilab/Программирование (дата обращения: 18.11.2020).

Химические науки

УЧАСТИЕ КЛЕТОЧНОЙ МЕМБРАНЫ В ОБМЕННЫХ ПРОЦЕССАХ

Кашуба К.Ю., Ковтун Е.С., Боровская Л.В.
Кубанский Государственный Технологический Университет, Краснодар,
e-mail: kashubaxenia@yandex.ru

Мембрана живой клетки (клеточная мембрана, плазмалемма, цитолемма, плазматическая мембрана) – тончайшая плёнка, состоящая из билипидного слоя, обладающая способностью совершать процесс обмена потоками энергии и продуктов метаболизма.

Клеточная теория появилась в семнадцатом веке, но только спустя 200 лет появилась полная теория о клеточных мембранах, которая подтвердила, что именно мембрана отделяет клетку от внешнего мира. Упорные исследования Морица Траумбе, определили, что некоторый внешний слой клетки является полупроницаемым, потому что должен обеспечивать перенос ионов. К сожалению, у учёного не было прямых доказательств состава этого внешнего слоя клетки. Современная теория строения клетки утверждает, что в билипидном слое каждая молекула фосфолипидов имеет гидрофильную голову и гидрофобный хвост. В плазмалемме они расположены головами наружу, а хвостами вовнутрь, что обеспечивают барьерную функцию. В би-слой погружено огромное множество молекул белков, которые, в свою очередь, выполняют ряд важнейших функций, важнейшими из которых являются:

– получения и превращения химических сигналов извне (рецепторы);

– транспорт ионов (ионные каналы);

– передача гормональных сигналов в клетку;

– ферментативная активность в преобразованиях веществ;

– межклеточный контакт, ведущий к образованию тканей и органов;

– избирательный транспорт веществ в клетку и из неё;

– адгезивная роль в связывание цитоскелета с внеклеточным матриксом.

Клеточная мембрана принимает участие в пиноцитозе и фагоцитозе, контролирует регуляцию водного баланса в клетках и выводит из них продукты жизнедеятельности.

Питание клетки – это процесс обеспечения живой клетки всеми незаменимыми питательными веществами для её же поддержания и образования новых единиц живого.

Транспорт ионов, потоков питательных веществ и продуктов- метаболитов через мембраны происходит с участием мембранных белков и электролитов, и движущей силой этого физико-химического процесса является изменение химического потенциала вместе с его электрохимической составляющей:

$$\mu' = \mu_0 + RTlnC + zF\varphi, \qquad (1)$$

где μ_0 – стандартный химический потенциал вещества, равный парциальной молярной энергии Гиббса,

F – $9{,}65 \cdot 10^4$ $\frac{\text{Кл}}{\text{моль}}$ число Фарадея;

z – заряд иона электролита (в элементарных единицах заряда);

φ – потенциал электрического поля, [В].

Механизм переноса веществ через мембраны может идти двумя путями: пассивный и активный переносы.

Пассивный транспорт – самопроизвольный процесс, перенесение вещества за счет градиента концентрации и выравнивания значений химических потенциалов, что сопровождается понижением энергии Гиббса, т.е. течение процесса проходит спонтанно, без расхода энергии. Здесь механизм описывается диффузией и плот-