**Практическая работа 7. Составление спецификации для разработки программного модуля**

# Цель занятия: получить практические навыки составления спецификации для разработки программного модуля

**Краткие теоретические сведения**

# Понятие спецификации

Под спецификацией понимается формальное описание функций и данных программы, с которыми эти функции оперируют. Различают видимые данные, т.е. входные и выходные параметры, а также скрытые данные, которые не привязаны к реализации и определяют интерфейс с другими функциями.

Предусловия – это ограничения на совокупность входных параметров и постусловия - ограничения на выходные параметры.

Спецификация программы должна иметь точное, однозначное и недвусмысленное описание программы с помощью математических понятий, терминов, правил синтаксиса и семантики языка спецификации. В языке спецификаций могут быть понятия и конструкции, которые нельзя выполнить на компьютере, они представляются последовательностью операций, функций, понятных для интерпретации. Спецификация программы должна содержать информацию, необходимую для построения алгоритма программы.

Спецификация содержит набор требований для программы в целом, позволяет зафиксировать детальные требования к разработке, указать роли и ответственности сторон, сроки и стоимость реализации. Так вы сможете четко понимать, что и когда будет реализовано, и, в случае, разногласий, иметь письменное подтверждение договоренностей.

Набор требований предназначен для того, чтобы установить базу для соглашения между заказчиком и разработчиком (или подрядчиками) о том, как должен функционировать программный продукт.

**Выделяют следующие необходимые свойства набора требований:**

* однократное упоминание отдельных требований;
* отсутствие пересечений между требованиями;
* явное указание связей между требованиями;
* полнота;
* непротиворечивость;
* определение ограничений, области действия и контекста для каждого требования;
* модифицируемость, конфигурируемость, удобство поддержки. Методика составления спецификаций требований к программному обеспечению (IEEE 830-1998) определяет содержание и качественные характеристики правильно составленной спецификации требований

# Примерный состав спецификации

* Наименование модуля
* Цель модуля
* Функциональные возможности модуля
* Взаимодействие с другими модулями
* Взаимодействие с пользователем
* Входные данные
* Выходные данные
* Навигация внутри модуля
* Требования к макету
* Обработка ошибок

**Пример спецификации:**

Наименование экранной формы: Главная страница приложения

Язык разработки: С#

Среда разработки: MSVisualStudio *номер*

Цель: cоздать головной модуль, позволяющий осуществить переход к выполнению других модулей программного продукта.

Входные данные: нажатия на кнопки (элементы навигации модуля).

Навигация:

* + во время запуска программы пользователь попадает на форму с навигацией, с помощью которой можно получить информацию о программе, либо запустить соответствующие модули.

Выходные параметры: запуск новой формы программного продукта с новым модулем Действия:

* + выбрать нужный элемент навигации;
  + нажать на нужный элемент навигации;
  + запустить нужный модуль с помощью элемента навигации.

Требования к макету страницы:

На форме должно быть три кнопки, которые находятся посередине формы.

Кнопки имеют название:

* + Пройти тест;
  + О программе;
  + Выход;

Шрифт всех кнопок TimesNewRoman, размер 14 пт. Кнопки зеленого цвета. Основной цвет формы – серый.

При нажатии на кнопку «Пройти тест», открывается модуль, с помощью которого необходимо зарегистрировать пользователя, после чего открывается новая форма с самим тестом.

При нажатии на кнопку «О программе» открывается новое окно с информацией о программе.

При нажатии на кнопку «Выход» все остальные окна формы закрываются.

Сообщения об ошибках: нет.

Алгоритм программы описать словами и предоставить блок-схему.

Прикладывается тест-кейсы для задачи

|  |  |
| --- | --- |
| Тестовый пример # | Уникальный номер |
| Приоритет тестирования | Высокий (функция критична для работоспособности всей системы)  Средний (если функция важна, но не критична для работоспособности всей системы)  Низкий (затрагивает малозначимый аспект функционирования системы, выполнение не влияет на общую работоспособность системы) |
| Заголовок/название теста | Заголовок теста |
| Краткое изложение теста | Что делает тест |
| Этапы теста | Пошаговое описание действий |
| Тестовые данные | Данные для тестирования |
| Ожидаемый результат | Что ожидается |
| Фактический результат | Что произошло |
| Статус | Успех |
| Предварительное условие | Условие для работы (программа запущена, установлен фреймворк, переход на страницу по url, …) |
| Постусловие | Система возвращается в прежнее состояние |
| Примечания/комментарии | Важно обрабатывать такие-то исключения…, чтобы избежать сбоев программы |

# Задание. Создайте спецификацию для разработки программы:

* 1. Вычислить периметр, площадь прямоугольного треугольника по данным катетам
  2. Вычислить периметр и площадь треугольника, используя длины сторон (a, b, c) по формуле Герона: S = (p \* (p-a) \* (p-b) \* (p-c))^1/2.
  3. Вычислить периметр, площадь прямоугольного треугольника по гипотенузе и катету
  4. Вычислить объем усеченного конуса по значениям высоты и радиусов оснований
  5. Вычислить объем и площадь поверхности куба по значению диагонали куба.
  6. Вычислить объем цилиндра по высоте и радиусу оснований
  7. Вычислить площадь поверхности правильной треугольной усеченной пирамиды по сторонам оснований и высоте.
  8. Вычислить объем правильной треугольной пирамиды по высоте и стороне основания.
  9. Вычислить объем шара по радиусу сечения, находящегося на заданном расстоянии от центра шара.
  10. Вычислить объем конуса по значению радиуса основания и образующей.
  11. Вычислить площадь круга по длине окружности.
  12. Вычислить площадь поверхности правильной шестиугольной призмы по высоте и длине стороны основания.
  13. Вычислить объем октаэдра по площади его грани.
  14. Вычислить площадь полной поверхности правильной четырехугольной пирамиды по высоте и стороне основания.
  15. Вычислить площадь поверхности прямоугольного параллелепипеда по значениям его измерений.
  16. Вычислить объем правильной треугольной усеченной пирамиды по сторонам оснований и высоте.
  17. Вычислить объем правильной четырехугольной призмы по площади диагонального сечения.
  18. Вычислить диагонали прямоугольного параллелепипеда по значениям его измерений.
  19. Вычислить периметр, площадь равностороннего треугольника
  20. Вычислить объем прямоугольного параллелепипеда по значениям его измерений.
  21. Вычислить объем конуса по значениям радиуса основания и площади осевого сечения.
  22. Вычислить объем шара по площади его большого круга.
  23. Вычислить объем правильной шестиугольной призмы по высоте и длине стороны основания.
  24. Найти координаты точки пересечения двух прямых на плоскости по уравнениям линий вида y=kx+b.
  25. Проверить, лежат ли три точки на одной прямой, используя их декартовы координаты (x1,y1), (x2,y2) и (x3,y3).
  26. Вычислить расстояние от точки до прямой на плоскости, если известны координаты точки и коэффициенты уравнения прямой (Ax+By+C=0).
  27. Рассчитать объём тетраэдра, заданного четырьмя вершинами в пространстве с координатами (x1,y1,z1), (x2,y2,z2), (x3,y3,z3), (x4,y4,z4).
  28. Определить длину отрезка на координатной оси по двум точкам с известными координатами.
  29. По известным длинам трёх сторон треугольника определить вид угла напротив наибольшей стороны (острый, тупой или прямой).
  30. Определить радиус вписанной окружности правильного многоугольника по числу сторон и длине каждой стороны.
  31. Найти точку пересечения медиан треугольника, если известны координаты вершин треугольника.
  32. Определить угол между двумя векторами на плоскости, зная их компоненты vec\_a (ax,ay) и vec\_b (bx,by).
  33. Вычислить расстояние между двумя параллельными плоскостями, заданными своими уравнениями общего вида Ax + By + Cz + D1 = 0 и Ax + By + Cz + D2 = 0.

# Контрольные вопросы

1. Что такое спецификация?
2. Назначение спецификации.
3. Свойства набора требований к ПП.
4. Примерный состав спецификации.
5. Где зафиксирована методика составления спецификаций к программному обеспечению?

# Используемая литература

Гниденко, И. Г. Технология разработки программного обеспечения: учеб. пособие для СПО / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — М.: Издательство Юрайт, 2017.

**Пример спецификации (фрагмент)**

**Исходные данные:**

Есть массив целых чисел размером n. С клавиатуры вводится два числа

- порядковые номера элементов массива, которые необходимо суммировать.

Например, если ввели 3 и 5 - суммируются 3-й и 5-й элементы. Описать функцию целого типа, возвращающую сумму заданных элементов массива.

# Решение

Наименование: Программа вычисления суммы двух заданных

элементов массива Язык разработки: С#

Среда разработки: MS Visual Studio *номер*

Цель: создать программу вычисления суммы заданных элементов массива.

Входные данные: Массив целых чисел, размер массива, номера элементов массива для суммирования.

Навигация: во время запуска программы на экране для пользователя появляются указания и подсказки для ввода данных.

Выходные параметры: Результаты вычислений или сообщения об ошибках

Навигация внутри модуля: Программа содержит встроенный метод для вычисления суммы заданных элементов массива Сообщения об ошибках:

- обработка ошибок ввода данных, если введено не число;

- ошибки выхода за пределы массива.

**Пример описания программы (алгоритм задачи)**

Задача: есть массив целых чисел размером n. С клавиатуры вводится два числа – порядковые номера элементов массива, которые необходимо суммировать. Например, если ввели 3 и 5 - суммируются 3-й и 5-й элементы. Описать функцию целого типа, возвращающую сумму заданных элементов массива.

Алгоритм решения:

Для решения задачи необходимо написать функцию, которая суммирует элементы массива по входным данным.

Основная программа: формируем массив. Вводим порядковые элементы массива, формируем результат суммы в новой переменной result, при этом вызываем созданную функцию. Выводим полученный результат на экран.

Подпрограмма: необходимо проверить, что введённые номера находятся в диапазоне допустимых значений (от 1 до n). Если хотя бы один номер выходит за пределы диапазона, то функция возвращает 0 (ошибку). Если проверка прошла успешно, берем элемент массива по первому номеру и элемент массива по второму номеру, суммируем их.

**Пример блок-схемы алгоритма**

Основная программа Подпрограмма

Изображение выглядит как текст, диаграмма, линия, снимок экрана

Контент, сгенерированный ИИ, может содержать ошибки. Изображение выглядит как текст, диаграмма, снимок экрана, линия

Контент, сгенерированный ИИ, может содержать ошибки.

**Примеры тест-кейсов**

|  |  |
| --- | --- |
| Тестовый пример # | t1 |
| Приоритет тестирования | Средний |
| Заголовок/название теста | Корректный подсчёт суммы элементов массива |
| Краткое изложение теста | Проверить правильную работу функции, осуществляющей суммирование элементов массива по введённым индексам |
| Этапы теста | 1. Ввести индексы элементов массива.  2. Выполнить проверку вычисленной суммы элементов. |
| Тестовые данные | индексы: i = 3, j = 5 |
| Ожидаемый результат | Вернуть сумму элементов массива по указанным индексам: сумма элементов массива под номерами 3 и 5 должна составить 80. |
| Фактический результат | Фактически было получено значение 80, совпадающее с ожидаемой суммой. |
| Статус | Успех |
| Предварительное условие | Программа загружена и готова к выполнению |
| Постусловие | Программа сохраняет своё рабочее состояние, никаких побочных эффектов нет |
| Примечания/комментарии | Используются стандартные целые числа и положительные индексы |

|  |  |
| --- | --- |
| Тестовый пример # | t2 |
| Приоритет тестирования | Высокий |
| Заголовок/название теста | Работа функции при индексах, превышающих размер массива |
| Краткое изложение теста | Проверить реакцию функции на неправильные индексы |
| Этапы теста | 1. Ввести индексы  2. Оценить поведение функции. |
| Тестовые данные | индексы: i = 6, j = 7 |
| Ожидаемый результат | Сообщение об ошибке («Неверные индексы») |
| Фактический результат | Отображается ошибка |
| Статус | Успех |
| Предварительное условие | Программа загружена и готова к выполнению |
| Постусловие | Программа сохраняет своё рабочее состояние, никаких побочных эффектов нет |
| Примечания/комментарии | Важно обработать ситуации выхода за пределы массива, чтобы избежать сбоев программы |