МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Филиал федерального государственного бюджетного образовательного учреждения высшего образования
«Кубанский государственный университет»
в г. Славянске-на-Кубани

УТВЕРЖДАЮ

Проректор по работе с филиалами
ФГБОУ ВО «Кубанский государственный университет»

_____ А.А. Евдокимов

«31» мая 2024 г.

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

МДК.01.01 РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ

специальность 09.02.07 Информационные системы и программирование

Краснодар 2024

Рабочая программа учебной дисциплины МДК.01.01 РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ разработана на основе Федерального государственного образовательного стандарта (далее – ФГОС) среднего профессионального образования (далее – СПО) по специальности 09.02.07 Информационные системы и программирование (технологический профиль), утвержденного приказом Министерства образования и науки Российской Федерации от «09» декабря 2016 г. № 1547, (зарегистрирован в Министерстве юстиции России 26.12.2016 г. рег. № 44936) и примерной основной образовательной программы по специальности 09.02.07 Информационные системы и программирование (утвержденная протоколом Федерального учебно-методического объединения по УГПС 09.00.00 от 15 июля 2021 г. №3).

Дисциплина	МДК.01.01 РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ	
Форма обучения	очная	
Учебный год	2024-2025	
3,4 курс	6 семестр	7 семестр
всего 344 часов, в том числе:		
лекции	60 ч.	100 ч.
практические занятия	58 ч.	114 ч.
самостоятельные занятия	–	–
консультация	–	–
промежуточная аттестация	–	12 ч.
форма итогового контроля	зачет	экзамен

Составитель: преподаватель _____ М.С. Бушуев

Утверждена на заседании предметной (цикловой) комиссии физико-математических дисциплин и специальных дисциплин УГС 09.00.00 Информатика и вычислительная техника протокол № 10 от «30» мая 2024 г.

Председатель предметной (цикловой) комиссии:

_____ М.С. Бушуев
«30» мая 2024 г.

Рецензенты:

Технический директор
ООО «Техностарт» _____ И.Г. Колодезный

Технический директор
ООО «ПРАЙ» _____ Б.А. Шишкин

ЛИСТ
согласования рабочей программы по учебной дисциплине
МДК.01.01 «Разработка программных модулей»

Специальность среднего профессионального образования:
09.02.07 Информационные системы и программирование

СОГЛАСОВАНО:

Нач. УМО филиала _____ А.С. Демченко
«31» мая 2024 г.

Заведующая библиотекой филиала _____ М.В. Фуфалько
«31» мая 2024 г.

Нач. ИВЦ (программно-информационное обеспечение образовательной программы) _____ В.А. Ткаченко
«31» мая 2024 г.

СОДЕРЖАНИЕ

1 ОБЩАЯ ХАРАКТЕРИСТИКА РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ МДК.01.01 РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ

1.1 Область применения программы

Рабочая программа учебной дисциплины МДК.01.01 «Разработка программных модулей» является частью основной профессиональной образовательной программой в соответствии с Федеральным государственным образовательным стандартом среднего профессионального образования (далее ФГОС СПО) и примерной основной образовательной программой для специальности 09.02.07 Информационные системы и программирование.

1.2 Место учебной дисциплины в структуре основной профессиональной образовательной программы

Дисциплина «Разработка программных модулей» относится к общепрофессиональным дисциплинам профессиональной подготовки.

1.3 Цели и задачи учебной дисциплины. Требования к результатам освоения учебной дисциплины

В результате освоения дисциплины обучающийся должен иметь практический опыт:

– в разработке кода программного продукта на основе готовой спецификации на уровне модуля;

– использовании инструментальных средств на этапе отладки программного продукта;

– проведении тестирования программного модуля по определенному сценарию;

– использовании инструментальных средств на этапе отладки программного продукта;

– разработке мобильных приложений.

В результате освоения дисциплины обучающийся должен уметь:

– осуществлять разработку кода программного модуля на языках низкого и высокого уровней;

– создавать программу по разработанному алгоритму как отдельный модуль;

– выполнять отладку и тестирование программы на уровне модуля;

– осуществлять разработку кода программного модуля на современных языках программирования;

– уметь выполнять оптимизацию и рефакторинг программного кода;

– оформлять документацию на программные средства.

В результате освоения дисциплины обучающийся должен *знать:*

– основные этапы разработки программного обеспечения;

– основные принципы технологии структурного и объектно-ориентированного программирования;

– способы оптимизации и приемы рефакторинга;

– основные принципы отладки и тестирования программных продуктов.

Максимальная учебная нагрузка обучающегося 344 часов, в том числе:

- обязательная аудиторная учебная нагрузка обучающегося 332 часа;
- промежуточная аттестация 12 часов.

1.4. Перечень планируемых результатов обучения по дисциплине (Перечень формируемых компетенций)

Освоение дисциплины «Разработка программных модуле» способствует формированию у студентов следующих профессиональных компетенций:

ПК 1.1. Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием

ПК 1.2 Разрабатывать программные модули в соответствии с техническим заданием

№ п.п.	Индекс компе-тенции	Содержание ком-петенции (или её части)	В результате изучения учебной дисциплины обучающиеся должны		
			знать	уметь	иметь практический опыт
6	ПК 1.1.	Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием	Основные этапы разработки программного обеспечения. Основные принципы технологии структурного и объектно-ориентированного программирования. Актуальная нормативно-правовая база в области документирования алгоритмов	Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием. Оформлять документацию на программные средства. Оценка сложности алгоритма	Разрабатывать алгоритм решения поставленной задачи и реализовывать его средствами автоматизированного проектирования.
7	ПК 1.2	Разрабатывать программные модули в соответствии с техническим заданием	Основные этапы разработки программного обеспечения. Основные принципы технологии структурного и объектно-ориентированного программирования. Знание API современных мобильных операционных систем.	Создавать программу по разработанному алгоритму как отдельный модуль. Оформлять документацию на программные средства. Осуществлять разработку кода программного модуля на языках низкого уровня и высокого уровней в том числе для мобильных платформ.	Разрабатывать код программного продукта на основе готовой спецификации на уровне модуля. Разрабатывать мобильные приложения

2 СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

2.1. Объем учебной дисциплины и виды учебной работы

Вид учебной работы	Всего часов	Семестр	
		6	7
Учебная нагрузка (всего)	344	118	226
Аудиторная нагрузка (всего)	332	118	214
в том числе:			
лекционные занятия	160	60	100
практические занятия	172	58	114
Самостоятельная работа			
в т.ч. консультации			
Промежуточная аттестация – экзамен	12	–	12

2.2 Структура дисциплины

Освоение учебной дисциплины МДК.01.01 РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ включает изучение следующих разделов и тем:

№	Тема	Всего часов	Лекции	Практические	Самостоятельная работа
		332	160	172	
1	***Тема 1.1.1 Жизненный цикл ПО***	6	6	–	
2	***Тема 1.1.2 Структурное программирование***	38	20	18	
3	***Тема 1.1.3Объектно-ориентированное программирование***	78	38	40	
4	***Тема 1.1.4Паттерн ы проектирования***	60	24	36	
5	***Тема 1.1.5. Событийно- управляемое программирование***	50	20	30	
6	***Тема 1.1.6 Оптимизация и рефакторинг кода***	32	16	16	
7	***Тема 1.1.7Разработка пользовательского интерфейса.***	32	16	16	
8	***Тема 1.1.8 Основы ADO.Net***	36	20	16	

2.3 Тематический план и содержание учебной дисциплины

МДК.01.01 РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ

Наименование разделов и тем профессионального модуля (ПМ), междисциплинарных курсов (МДК)	Содержание учебного материала, лабораторные работы и практические занятия	Объем в
		Программист
Тема 1.1.1 Жизненный цикл ПО	***Содержание***	**6**
	1. Понятие ЖЦ ПО. Этапы ЖЦ ПО.	
Тема 1.1.2 Структурное программирование	***Содержание***	**38**
	1. Технология структурного программирования.	

	2. Инструментальные средства оформления и документирования алгоритмов программ	
	3. Оценка сложности алгоритма: классификация, классы алгоритмов, неразрешимые задачи	
	В том числе практических занятий и лабораторных работ	18
	1. Оценка сложности алгоритмов сортировки.	
	2. Оценка сложности алгоритмов поиска.	
	3. Оценка сложности рекурсивных алгоритмов.	
	4. Оценка сложности эвристических алгоритмов.	
Тема 1.1.3Объектно-ориентированное программирование	***Содержание***	**78**
	1. Основные принципы объектно-ориентированного программирования. Классы: основные понятия.	
	2. Перегрузка методов.	
	3. Операции класса.	
	4. Иерархия классов.	
	5. Синтаксис интерфейсов.	
	6. Интерфейсы и наследование.	
	7. Структуры.	
	8. Делегаты.	
	9. Регулярные выражения	
	10 Коллекции. Параметризованные классы.	
	11 Указатели	
	12 Операции со списками	
	В том числе практических занятий и лабораторных работ	*40*
	1. Работа с классами.	
	2. Перегрузка методов.	
	3. Определение операций в классе.	
	4. Создание наследованных классов	
	5. Работа с объектами через интерфейсы.	
	6. Использование стандартных интерфейсов.	
	7. Работа с типом данных структура.	
	8. Коллекции. Параметризованные классы.	
	9. Использование регулярных выражений	
	10. Операции со списками.	
Тема 1.1.4Паттерн ы проектирования	***Содержание***	**60**
	1. Назначение и виды паттернов.	
	2. Основные шаблоны.	
	3. Порождающие шаблоны.	
	4. Структурные шаблоны.	
	5. Поведенческие шаблоны.	
	В том числе практических занятий и лабораторных работ	36
	1. Использование основных шаблонов.	
	2. Использование порождающих шаблонов.	
	3. Использование структурных шаблонов.	
	4. Использование поведенческих шаблонов.	
Тема 1.1.5. Событийно- управляемое программирование	***Содержание***	**50**
	1. Событийно-управляемое программирование	
	2. Элементы управления. Диалоговые окна. Обработчики событий.	
	3. Введение в графику	
	В том числе практических занятий и лабораторных работ	30

	1. Разработка приложения с использованием текстовых компонентов	
	2. Разработка приложения с несколькими формами.	
	3. Разработка приложения с не визуальными компонентами.	
	4. Разработка игрового приложения.	
	5. Разработка приложения с анимацией.	
Тема 1.1.6 Оптимизация и рефакторинг кода	***Содержание***	**32**
	1. Методы оптимизации программного кода.	
	2. Цели и методы рефакторинга.	
	В том числе практических занятий и лабораторных работ	16
	1. Оптимизация и рефакторинг кода.	
Тема 1.1.7Разработка пользовательского интерфейса.	***Содержание***	**32**
	1. Правила разработки интерфейсов пользователя.	
	В том числе практических занятий и лабораторных работ	16
	1. Разработка интерфейса пользователя.	
Тема 1.1.8 Основы ADO.Net	***Содержание***	**36**
	1. Работа с базами данных	
	2. Доступ к данным	
	3. Создание таблицы, работа с записями.	
	4. Способы создания команд	
	В том числе практических занятий и лабораторных работ	**16**
	1. Создание приложения с БД	
	2. Создание запросов к БД	
	3. Создание хранимых процедур	
	Итоговая аттестация	**12**
	Всего	**344**

2.4 Содержание разделов дисциплины

2.4.1 Занятия лекционного типа

№ п/п	Наименование раздела	Содержание раздела	Форма текущего контроля
1	***Тема 1.1.1 Жизненный цикл ПО***	Понятие ЖЦ ПО. Этапы ЖЦ ПО.	У, КР
2	***Тема 1.1.2 Структурное программирование***	1. Технология структурного программирования. 2. Инструментальные средства оформления и документирования алгоритмов программ 3. Оценка сложности алгоритма: классификация, классы алгоритмов, неразрешимые задачи	У, КР
3	***Тема 1.1.3Объектно-ориентированное программирование***	1. Основные принципы объектно-ориентированного программирования. Классы: основные понятия. 2. Перегрузка методов. 3. Операции класса. 4. Иерархия классов. 5. Синтаксис интерфейсов. 6. Интерфейсы и наследование. 7. Структуры. 8. Делегаты. 9. Регулярные выражения 10. Коллекции. Параметризованные классы. 11. Указатели 12. Операции со списками	У, КР
4	***Тема 1.1.4Паттерн ы проектирования***	1. Назначение и виды паттернов. 2. Основные шаблоны. 3. Порождающие шаблоны. 4. Структурные шаблоны. 5. Поведенческие шаблоны.	У, КР
5	***Тема 1.1.5. Событийно- управляемое программирование***	1. Событийно-управляемое программирование 2. Элементы управления. Диалоговые окна. Обработчики событий. 3. Введение в графику	У, КР
6	***Тема 1.1.6 Оптимизация и рефакторинг кода***	1. Методы оптимизации программного кода. 2. Цели и методы рефакторинга.	У, КР
7	***Тема 1.1.7Разработка пользовательского интерфейса.***	1. Правила разработки интерфейсов пользователя	У, КР
8	***Тема 1.1.8 Основы ADO.Net***	1. Работа с базами данных 2. Доступ к данным 3. Создание таблицы, работа с записями. 4. Способы создания команд	У, КР

Примечание: Р - написание реферата, У - устный опрос, КР - контрольная работа

2.4.2 Занятия семинарского типа

- не предусмотрены

2.4.3 Практические занятия (Лабораторные занятия)

№ п/п	Наименование раздела	Содержание раздела	Форма текущего контроля
1	***Тема 1.1.1 Жизненный цикл ПО***	–	–
2	***Тема 1.1.2 Структурное программирование***	1. Оценка сложности алгоритмов сортировки. 2. Оценка сложности алгоритмов поиска. ***3.*** Оценка сложности рекурсивных алгоритмов. 4. Оценка сложности эвристических алгоритмов.	ПР
3	***Тема 1.1.3Объектно-ориентированное программирование***	1. Работа с классами. 2. Перегрузка методов. 3. Определение операций в классе. 4. Создание наследованных классов 5. Работа с объектами через интерфейсы. 6. Использование стандартных интерфейсов. 7. Работа с типом данных структура. 8. Коллекции. Параметризованные классы. 9. Использование регулярных выражений 10. Операции со списками.	ПР
4	***Тема 1.1.4Паттерн ы проектирования***	1. Использование основных шаблонов. 2. Использование порождающих шаблонов. 3. Использование структурных шаблонов. 4. Использование поведенческих шаблонов.	ПР
5	***Тема 1.1.5. Событийно- управляемое программирование***	1. Разработка приложения с использованием текстовых компонентов 2. Разработка приложения с несколькими формами. 3. Разработка приложения с не визуальными компонентами. 4. Разработка игрового приложения. 5. Разработка приложения с анимацией.	ПР
6	***Тема 1.1.6 Оптимизация и рефакторинг кода***	1. Оптимизация и рефакторинг кода.	ПР
7	***Тема 1.1.7Разработка пользовательского интерфейса.***	1. Разработка интерфейса пользователя.	ПР
8	***Тема 1.1.8 Основы ADO.Net***	1. Создание приложения с БД 2. Создание запросов к БД ***3.*** Создание хранимых процедур	ПР

Примечание: ПР- практическая работа

2.4.4 Содержание самостоятельной работы

Не предусмотрено

2.4.5 Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине

Не предусмотрено

3 ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

Для освоения курса «Разработка программных модулей» предусматривается использование в учебном процессе активных и интерактивных форм проведения аудиторных и внеаудиторных занятий с целью формирования и развития профессиональных навыков обучающихся.

В процессе обучения применяются образовательные технологии личностно-деятельностного, развивающего и проблемного обучения. Обязателен лабораторный практикум по разделам дисциплины.

В учебном процессе наряду с традиционными образовательными технологиями используются компьютерное тестирование, тематические презентации, интерактивные технологии.

3.1. Образовательные технологии при проведении лекций

Тема	Виды применяемых образовательных технологий	Кол. час
Тема 1.1.1 Жизненный цикл ПО	Аудиовизуальная технология, технология развивающего обучения	6
Тема 1.1.2 Структурное программирование	Аудиовизуальная технология, личностно-деятельностное обучение	20
Тема 1.1.3Объектно- ориентированное программирование	Аудиовизуальная технология, личностно-деятельностное обучение	38
Тема 1.1.4Паттерн ы проектирования	Аудиовизуальная технология, дифференцированное обучение	24
Тема 1.1.5. Событийно- управляемое программирование	Аудиовизуальная технология, технология развивающего обучения	20
Тема 1.1.6 Оптимизация и рефакторинг кода	Аудиовизуальная технология, дифференцированное обучение	16
Тема 1.1.7Разработка пользовательского интерфейса.	Аудиовизуальная технология, технология развивающего обучения	16
Тема 1.1.8 Основы ADO.Net	Аудиовизуальная технология, технология развивающего обучения	20
	Итого	160

3.2. Образовательные технологии при проведении практических занятий

Тема	Виды применяемых образовательных технологий	Кол. час
Тема 1.1.1 Жизненный цикл ПО	Технология личностно-деятельностного обучения, технология проблемного обучения	–
Тема 1.1.2 Структурное программирование	Технология личностно-деятельностного обучения, технология проблемного обучения	18
Тема 1.1.3Объектно- ориентированное программирование	Технология проблемного обучения	40
Тема 1.1.4Паттерн ы проектирования	Технология личностно-деятельностного обучения, технология проблемного обучения	36
Тема 1.1.5. Событийно- управляемое программирование	Технология личностно-деятельностного обучения, технология проблемного обучения	30
Тема 1.1.6 Оптимизация и рефакторинг кода	Технология личностно-деятельностного обучения, технология проблемного обучения	16
Тема 1.1.7Разработка пользовательского интерфейса.	Технология личностно-деятельностного обучения, технология проблемного обучения	16

Тема 1.1.8 Основы ADO.Net	Технология личностно-деятельностного обучения, технология проблемного обучения	16
	Итого	172

4 УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ МДК.01.01 РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ

4.1 Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине

Для реализации программы учебной дисциплины должны быть предусмотрены следующие специальные помещения:

Лаборатории Программного обеспечения и сопровождения компьютерных систем, оснащенные в соответствии с п. 6.2.1. Примерной программы по специальности:

Оснащенные базы практики, в соответствии с п 6.2.3 Примерной программы по специальности.

4.2 Перечень необходимого программного обеспечения

1. 7-zip (лицензия на англ. http://www.7-zip.org/license.txt)
2. Adobe Acrobat Reade (лицензия - https://get.adobe.com/reader/?loc=ru&promoid=KLXME)
3. Adobe Flash Player (лицензия - https://get.adobe.com/reader/?loc=ru&promoid=KLXME)
4. Apache Open Office (лицензия - http://www.openoffice.org/license.html)
5. Free Commander (лицензия - https://freecommander.com/ru/%d0%bb%d0%b8%d1%86%d0%b5%d0%bd%d0%b7%d0%b8%d1%8f/)
6. Google Chrome (лицензия - https://www.google.ru/chrome/browser/privacy/eula_text.html)
7. LibreOffice (в свободном доступе)
8. Mozilla Firefox (лицензия - https://www.mozilla.org/en-US/MPL/2.0/)
9. Oracle VM VirtualBox (лицензия - https://www.virtualbox.org/wiki/GPL)

5 ПЕРЕЧЕНЬ ОСНОВНОЙ И ДОПОЛНИТЕЛЬНОЙ УЧЕБНОЙ ЛИТЕРАТУРЫ, НЕОБХОДИМОЙ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ

5.1 Основная литература

1. Белугина, С. В. Разработка программных модулей программного обеспечения для компьютерных систем. Прикладное программирование : учебное пособие / С. В. Белугина. — Санкт-Петербург : Лань, 2020. — 312 с. — ISBN 978-5-8114-4496-0. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/133920. — Режим доступа: для авториз. пользователей.

5.2 Дополнительная литература

1. Маран, М. М. Программная инженерия : учебное пособие / М. М. Маран. — Санкт-Петербург : Лань, 2021. — 196 с. — ISBN 978-5-8114-3032-1. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/169168. — Режим доступа: для авториз. пользователей.

2. Федорова, Г. Н. Разработка, внедрение и адаптация программного обеспечения отраслевой направленности : учебное пособие / Г. Н. Федорова. — Москва : КУРС : ИНФРА-М, 2021. — 336 с. — (Среднее профессиональное образование). - ISBN 978-5-906818-41-6. - Текст : электронный. - URL: https://znanium.com/catalog/product/1138896. – Режим доступа: по подписке.

3. Мартишин, С. А. Базы данных. Практическое применение СУБД SQL- и NoSQL-типа для проектирования информационных систем : учебное пособие / С.А. Мартишин, В.Л. Симонов, М.В. Храпченко. — Москва : ФОРУМ : ИНФРА-М, 2021. — 368 с. — (Среднее профессиональное образование). - ISBN 978-5-8199-0785-6. - Текст : электронный. - URL: https://znanium.com/catalog/product/1243192. – Режим доступа: по подписке.

5.3 Периодические издания

1. Computerworld Россия. – URL: http://dlib.eastview.com/browse/publication/64081/udb/2071.
2. Windows IT Pro / Re. – URL: http://dlib.eastview.com/browse/publication/64079/udb/2071.
3. БИТ. Бизнес & информационные технологии – URL : http://dlib.eastview.com/browse/publication/66752/udb/2071.
4. Виртуализация. Облачные структуры. Системы хранения данных. – URL : https://dlib.eastview.com/browse/publication/84826/udb/2071.
5. Информатика, вычислительная техника и инженерное образование. - URL: https://www.elibrary.ru/title_about.asp?id=32586.
6. Информационно-управляющие системы. – URL: http://dlib.eastview.com/browse/publication/71235.
7. Мир больших данных. – URL : https://dlib.eastview.com/browse/publication/90728/udb/2071.
8. Мир ПК. – URL: http://dlib.eastview.com/browse/publication/64067/udb/2071.
9. Прикладная информатика. – URL: https://e.lanbook.com/journal/2067#journal_name.
10. Программные продукты и системы. – URL: https://dlib.eastview.com/browse/publication/64086/udb/2071.
11. Программные продукты и системы. – URL: http://dlib.eastview.com/browse/publication/64086/udb/2071.

12. САПР и графика. - URL: https://sapr.ru/list,

13. Системный администратор. – URL: https://dlib.eastview.com/browse/publication/66751/udb/2071.

14. Системный анализ и прикладная информатика. – URL: https://e.lanbook.com/journal/2420#journal_name.

5.4 Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

1. ЭБС «BOOK.ru» [учебные издания – коллекция для СПО] : сайт. – URL: https://www.book.ru/cat/576.

2. ЭБС «Университетская библиотека ONLINE» [учебные, научные здания, первоисточники, художественные произведения различных издательств; журналы; мультимедийная коллекция, карты, онлайн-энциклопедии, словари] : сайт. – URL: http://biblioclub.ru/index.php?page=main_ub_red.

3. ЭБС издательства «Лань» [учебные, научные издания, первоисточники, художественные произведения различных издательств; журналы] : сайт. – URL: http://e.lanbook.com.

4. ЭБС «Юрайт» [учебники и учебные пособия издательства «Юрайт»] : сайт. – URL: https://urait.ru/.

5. ЭБС «Znanium.com» [учебные, научные, научно-популярные материалы различных издательств, журналы] : сайт. – URL: http://znanium.com/.

6. Научная электронная библиотека. Монографии, изданные в издательстве Российской Академии Естествознания [полнотекстовый ресурс свободного доступа] : сайт. – URL: https://www.monographies.ru/.

7. Научная электронная библиотека статей и публикаций «eLibrary.ru» [российский информационно-аналитический портал в области науки, технологии, медицины, образования; большая часть изданий – свободного доступа] : сайт. – URL: http://elibrary.ru.

8. Базы данных компании «Ист Вью» [периодические издания (на русском языке)] : сайт. – URL: http://dlib.eastview.com.

9. Российская электронная школа : государственная образовательная платформа [полный школьный курс уроков] : сайт. – URL: https://resh.edu.ru/.

10. Единое окно доступа к образовательным ресурсам : федеральная информационная система свободного доступа к интегральному каталогу образовательных интернет-ресурсов и к электронной библиотеке учебно-методических материалов для всех уровней образования: дошкольное, общее, среднее профессиональное, высшее, дополнительное : сайт. – URL: http://window.edu.ru.

11. Федеральный центр информационно-образовательных ресурсов [для общего, среднего профессионального, дополнительного образования; полнотекстовый ресурс свободного доступа] : сайт. – URL: http://fcior.edu.ru.

12. Единая коллекция цифровых образовательных ресурсов [для преподавания и изучения учебных дисциплин начального общего, основного общего и среднего (полного) общего образования; полнотекстовый ресурс свободного доступа] : сайт. – URL: http://school-collection.edu.ru.

13. Официальный интернет-портал правовой информации. Государственная система правовой информации [полнотекстовый ресурс свободного доступа] : сайт. – URL: http://publication.pravo.gov.ru.

14. Кодексы и законы РФ. Правовая справочно-консультационная система [полнотекстовый ресурс свободного доступа] : сайт. – URL: http://kodeks.systecs.ru.

15. ГРАМОТА.РУ : справочно-информационный интернет-портал : сайт. – URL: http://www.gramota.ru.

16. Энциклопедиум [Энциклопедии. Словари. Справочники : полнотекстовый ресурс свободного доступа] // ЭБС «Университетская библиотека ONLINE» : сайт. – URL: http://enc.biblioclub.ru/.

17. СЛОВАРИ.РУ. Лингвистика в Интернете : лингвистический портал : сайт. – URL: http://slovari.ru/start.aspx?s=0&p=3050.

18. Электронный каталог Кубанского государственного университета и филиалов. – URL: http://212.192.134.46/MegaPro/Web/Home/About.

6. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ОБУЧАЮЩИМСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Учащиеся для полноценного освоения курса «Разработка программных модулей» должны составлять конспекты как при прослушивании его теоретической (лекционной) части, так и при подготовке к практическим (семинарским) занятиям. Желательно, чтобы конспекты лекций и семинаров записывались в логической последовательности изучения курса и содержались в одной тетради. Это обеспечит более полную подготовку как к текущим учебным занятиям, так и сессионному контролю знаний.

Самостоятельная работа учащихся является важнейшей формой учебно- познавательного процесса. Цель заданий для самостоятельной работы - закрепить и расширить знания, умения, навыки, приобретенные в результате изучения дисциплины «Разработка программных модулей»; овладеть умением использовать полученные знания в практической работе; получить первичные навыки профессиональной деятельности.

Задания для самостоятельной работы выполняются в письменном виде во внеаудиторное время. Работа должна носить творческий характер, при ее оценке преподаватель в первую очередь оценивает обоснованность и оригинальность выводов. В письменной работе по теме задания учащийся должен полно и всесторонне рассмотреть все аспекты темы, четко сформулировать и аргументировать свою позицию по исследуемым вопросам. Выбор конкретного задания для самостоятельной работы проводит преподаватель, ведущий практические занятия в соответствии с перечнем, указанным в планах практических занятий.

Отчеты по практическим занятиям должны содержать полные ответы на поставленные задания, необходимые таблицы должны быть заполнены.

Общие правила выполнения письменных работ

На первом занятии студенты должны быть проинформированы о необходимости соблюдения норм академической этики и авторских прав в ходе обучения. В частности, предоставляются сведения:

- общая информация об авторских правах;
- правила цитирования;
- правила оформления ссылок;

Все имеющиеся в тексте сноски тщательно выверяются и снабжаются «адресами».

Недопустимо включать в свою работу выдержки из работ других авторов без указания на это, пересказывать чужую работу близко к тексту без отсылки к ней, использовать чужие идеи без указания первоисточников (это касается и информации, найденной в Интернете). Все случаи плагиата должны быть исключены.

Список использованной литературы должен включать все источники информации, изученные и проработанные студентом в процессе выполнения работы, и должен быть составлен в соответствии с ГОСТ Р 7.0.5-2008 «Библиографическая ссылка. Общие требования и правила».

7 КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ «МДК.01.01 РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ»

7.1 Паспорт фонда оценочных средств

№ п/п	Контролируемые разделы (темы) дисциплины	Компетенции	Наименование оценочного средства
1.	***Тема 1.1.1 Жизненный цикл ПО***	ПК 1.1, ПК 1.2	Проверка конспектов, устный опрос,
2.	***Тема 1.1.2 Структурное программирование***	ПК 1.1, ПК 1.2	Проверка конспектов, тест
3.	***Тема 1.1.3Объектно- ориентированное программирование***	ПК 1.1, ПК 1.2	Проверка конспектов, тест
4.	***Тема 1.1.4Паттерн ы проектирования***	ПК 1.1, ПК 1.2	Проверка конспектов, тест
5.	***Тема 1.1.5. Событийно- управляемое программирование***	ПК 1.1, ПК 1.2	Проверка конспектов, тест
6.	***Тема 1.1.6 Оптимизация и рефакторинг кода***	ПК 1.1, ПК 1.2	устный опрос, тест
7.	***Тема 1.1.7Разработка пользовательского интерфейса.***	ПК 1.1, ПК 1.2	Проверка конспектов, устный опрос, тест
8	***Тема 1.1.8 Основы ADO.Net***	ПК 1.1, ПК 1.2	Проверка конспектов, устный опрос, тест

7.2 Критерии оценки знаний

Контроль и оценка результатов освоения учебной дисциплины осуществляется преподавателем в процессе проведения практических работ, тестирования, собеседования по результатам выполнения лабораторных работ, а также решения задач, составления рабочих таблиц и подготовки сообщений к уроку. Знания студентов на практических занятиях оцениваются отметками «отлично», «хорошо», «удовлетворительно» и «неудовлетворительно».

Код и наименование профессиональных и общих компетенций, формируемых в рамках модуля	Критерии оценки	Методы оценки
Раздел модуля 1. Анализ и проектирование программных решений		
ПК 1.1 Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием	Оценка «**отлично**» - техническое задание проанализировано, алгоритм разработан, соответствует техническому заданию и оформлен в соответствии со стандартами, пояснены его основные структуры. Указаны использованные стандарты в области документирования; выполнена оценка сложности алгоритма	Экзамен/зачет в форме собеседования: практическое задание по построению алгоритма в соответствии с техническим заданием

	Оценка «**хорошо**» -алгоритм разработан, оформлен в соответствии со стандартами и соответствует заданию, пояснены его основные структуры. Выполнена оценка сложности алгоритма Оценка «**удовлетворительно**» - алгоритм разработан и соответствует заданию.	Защита отчетов по практическим и лабораторным работам
ПК 1.2 Разрабатывать программные модули в соответствии с техническим заданием	Оценка «отлично» - программный разработан по имеющемуся алгоритму в среде разработки на указанном языке программирования методами объектно-ориентированного/ структурного программирования и полностью соответствует техническому заданию, соблюдены и пояснены основные этапы разработки; документация на модуль оформлена и соответствует стандартам. Оценка «хорошо» - программный модуль разработан по имеющемуся алгоритму в среде разработки (на указанном языке программирования) методами объектно- ориентированного/ структурного программирования и практически соответствует техническому заданию с незначительными отклонениями, пояснены основные этапы разработки; документация на модуль оформлена и соответствует стандартам. Оценка «удовлетворительно» - программный модуль разработан по имеющемуся алгоритму в среде разработки (на указанном языке программирования) методами объектно-ориентированного/ структурного программирования и соответствует техническому заданию; документация на модуль оформлена без существенных отклонений от стандартов.	Экзамен/зачет в форме собеседования: практическое задание по разработке программного модуля в соответствии с техническим заданием Защита отчетов по практическим и лабораторным работам Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики

7.3 Оценочные средства для проведения текущей аттестации

-фронтальный опрос

-индивидуальный устный опрос

-письменный контроль

-тестирование по теоретическому материалу

-практическая (лабораторная) работа

Форма аттестации	Знания	Умения	Владения (навыки)	Личные качества студента	Примеры оценочных средств
Устный (письменный) опрос по темам	Контроль знаний по определенным проблемам	Оценка умения различать конкретные понятия	Оценка навыков работы с литературными источниками	Оценка способности оперативно и качественно отвечать на поставленные вопросы	Контрольные вопросы по темам прилагаются
Практические (лабораторные) работы	Контроль знания теоретических основ информатики и информационных технологий, возможностей и принципов использования современной компьютерной техники.	Оценка умения работать с современной компьютерной техникой, использовать возможности вычислительной техники и программного обеспечения при решении практических задач.	Оценка навыков работы с вычислительной техникой, прикладными программными средствами	Оценка способности оперативно и качественно решать поставленные на практических работах задачи и аргументировать результаты	Темы работ прилагаются
Тестирование	Контроль знаний по определенным проблемам	оценка умения различать некоторые понятия	Оценка навыков логического анализа и синтеза при сопоставлении некоторых понятий	Оценка способности оперативно и качественно отвечать на поставленные вопросы	Вопросы прилагаются

Контрольная работа. Контрольная работа является набором практических заданий и задач по темам изучаемой дисциплины, позволяющих формировать знания, а также умения обучающихся в области архитектуры аппаратных средств.

Примеры задач и вопросов к контрольной работе:

1. Технология структурного программирования.
2. Инструментальные средства оформления и документирования алгоритмов программ
3. Оценка сложности алгоритма: классификация, классы алгоритмов, неразрешимые задачи
4. Основные принципы объектно-ориентированного программирования. Классы: основные понятия.
5. Перегрузка методов.
6. Операции класса.
7. Иерархия классов.
8. Синтаксис интерфейсов.
9. Интерфейсы и наследование.

10. Структуры.
11. Делегаты.
12. Регулярные выражения
13. Коллекции. Параметризованные классы.
14. Указатели
15. Операции со списками
16. Назначение и виды паттернов.
17. Основные шаблоны.
18. Порождающие шаблоны.
19. Структурные шаблоны.
20. Поведенческие шаблоны.
21. Событийно-управляемое программирование
22. Элементы управления. Диалоговые окна. Обработчики событий.
23. Введение в графику
24. Методы оптимизации программного кода.
25. Цели и методы рефакторинга.
26. Правила разработки интерфейсов пользователя
27. Работа с базами данных
28. Доступ к данным
29. Создание таблицы, работа с записями.
30. Способы создания команд

Тест. Тест представляет собой систему стандартизированных заданий, позволяющих автоматизировать процедуру измерения уровня знаний обучающихся.

1. PYTHON является:
    a. Машинно - ориентированным языком (низкого уровня)
    b. Языком высокого уровня
    c. Объектно - ориентированным языком
2. Область применения PYTHON:
    a. Робототехника и искусственный интеллект
    b. Обучение
    c. Интернет
3. Год разработки PYTHON:
    a. 1990
    b. 1991
    c. 1993
4. Чувствителен ли PYTHON к регистру (большая или маленькая буквы):
    a. Да
    b. Нет
5. Какие существуют типы переменных (выбрать несколько выриантов):
    a. float
    b. list
    c. num
    d. int

e. bool
f. integer

6. Переменная int:
   a. вещественная переменная
   b. символьная строка
   c. логическая переменная
   d. целая переменная
7. Переменная str:
   a. символьная строка
   b. логическая переменная
   c. целая переменная
8. Переменная float:
   a. целая переменная
   b. вещественная переменная
   c. логическая переменная
9. Каков будет результат выполнения int("88"):
   a. "88"
   b. 88
   c. 88.00
10. Каков будет результат выполнения str(88):
   a. "88"
   b. 88
   c. 88.00
11. Имена переменных не могут включать:
   a. Русские буквы
   b. Латинские буквы
   c. Пробелы
   d. Скобки, знаки + = ! ? и др.
12. Какие имена являются правильными в PYTHON (выбрать несколько):
   a. N
   b. ABC
   c. sum
   d. 41And
   e. A+B
   f. _mam
13. Что будет в результате выполнения команды:
   a = 20
   b = a + 5
   *a = b * 100print(a)*
   a. 25
   b. 2500
   c. 25000
   d. 1000
14. Что будет в результате следующего действия *print(2**20)*
   a. 104576

b. 1048576

c. 964

d. 2

15. Что будет в результате выполнения следующего действия *print(23 % 2)*
    a. 11
    b. 1
    c. 0
16. Результатом вычисления *print(24 // 3)* будет число:
    a. 4
    b. 8
    c. 12
17. Что будет результатом выполнения алгоритма:
    Входные данные: a = 5, b = 7

    a = int(input())b = int(input())s = a + b print(s)
    a. 57
    b. 12
    c. 35
18. Что будет результатом выполнения алгоритма:
    Входные данные: a = 5, b = 7

    a = input()b = input()s = a + b print(s)
    a. 12
    b. 57
    c. 35
19. Что будет в результате выполнения следующего алгоритма:
    Входные данные: -57

    x = int(input())if x > 0:
    print(x)
    else:
    print(-x)
    a. -57

    b. 57
    c. 0
    d. -1
20. Что будет в результате выполнения программы:
    Входные данные: a = 10, b = 20

    a = int(input())b = int(input())if a < b:
    print(a)else:
    print(b)
    a. 10
    b. 20
    c. 30
    d. -10

21. Какой ряд чисел может образоваться после выполнения следующего алгоритма:
    for i in range(1,10):print(i)
    a. 1 2 3 4 5 6 7 8 9 10

b. 1 2 3 4 5 6 7 8 9

c. 0

22. Какой ряд чисел может образоваться после выполнения алгоритма:
for i in range(1,10+1):
print(i)
    a. 1 2 3 4 5 6 7 8 9 10
    b. 1 2 3 4 5 6 7 8 9 10 11
    c. 1 4 9 16
23. Что выведет программа после выполнения данного алгоритма:
Входные данные: name = Иванов

print('Как Ваша фамилия?')
name = input()
print('Здравствуйте,
'+ name + '!')
    a. Как Ваша фамилия? Здравствуйте, Иванов!
    b. Как Ваша фамилия? Здравствуйте, Иванов !
    c. Как Ваша фамилия? Здравствуйте, Иванов !
24. Как обозначается логический оператор И, ИЛИ, НЕ в питоне:
    a. OR, NOT, IF
    b. AND, OR, NOT
    c. AND, OR, IF
    d. AND, ELSE, NOT
25. Что будет в результате выполнения следующего алгоритма программы:
Входные данные: a = 15, b = 45

a = int(input())b = int(input())
if a % 10 == 0 or b % 10 == 0:print('YES')
else:
  print('NO')
    a. YES
    b. NO
26. Как будет записано число 18 после выполнения следующего алгоритма:
x = float(input())print(x)
    a. 18
    b. 18.0
    c. 18.00
27. Что будет после выполнения следующего листинга программы:
*for i in range(4)print(i) print(i ** 2)*
    a. 0 0 1 1 3 3 4 4
    b. 0 0 1 1 2 4 3 4
    c. 0 0 1 1 2 3 3 9
28. Результатом выполнения алгоритма цикла while будет:
i = 1
*while i <= 10:print(i ** 2)i = i + 1*
    a. 1 2 4 8 12 14

b. 1 2 16 24 32

c. 1 2 4 16 25 36 49 64 81 100

7.4 Оценочные средства для проведения промежуточной аттестации

Форма аттестации	Знания	Умения	Владение (навыки)	Личные качества студента	Примеры оценочных средств
Итоговая аттестация					
Экзамен	Контроль знания базовых положений в области операционных систем	Оценка умения понимать специальную терминологию	Оценка навыков логического сопоставления и характеристики объектов, работы и администрирования операционной системы	Оценка способности грамотно и четко излагать материал	Вопросы: прилагаются
		Оценка умения решать типовые задачи в области операционных систем	Оценка навыков логического мышления при решении задач в области операционных систем	Оценка способности грамотно и четко излагать ход решения задач в области архитектуры операционных систем и аргументировать результаты	Задачи прилагаются

7.4.1 Примерные вопросы для проведения промежуточной аттестации

1. Технология структурного программирования.
2. Инструментальные средства оформления и документирования алгоритмов программ
3. Оценка сложности алгоритма: классификация, классы алгоритмов, неразрешимые задачи
4. Основные принципы объектно-ориентированного программирования. Классы: основные понятия.
5. Перегрузка методов.
6. Операции класса.
7. Иерархия классов.
8. Синтаксис интерфейсов.
9. Интерфейсы и наследование.
10. Структуры.
11. Делегаты.
12. Регулярные выражения
13. Коллекции. Параметризованные классы.
14. Указатели
15. Операции со списками
16. Назначение и виды паттернов.
17. Основные шаблоны.
18. Порождающие шаблоны.

19. Структурные шаблоны.
20. Поведенческие шаблоны.
21. Событийно-управляемое программирование
22. Элементы управления. Диалоговые окна. Обработчики событий.
23. Введение в графику
24. Методы оптимизации программного кода.
25. Цели и методы рефакторинга.
26. Правила разработки интерфейсов пользователя
27. Работа с базами данных
28. Доступ к данным
29. Создание таблицы, работа с записями.
30. Способы создания команд

7.4.2 Примерные задачи для проведения промежуточной аттестации

1. Разработка простого калькулятора на Python
2. Разработка игры в города на Python
3. Разработка экспертной системы оценки стоимости ремонта после ДТП на Python
4. Разработка игры "Кто хочет стать миллионером" на Python
5. Разработка модели чат-бота на Python
6. Разработка игры "Дуэль рыцарей" на Python
7. Разработка конвертера валют на Python
8. Разработка утилиты для поиска файла по названию
9. Разработка конвертера мер и весов на Python
10. Разработка программы-теста по теме "Системное программирование" на Python
11. Разработка игры "Дуэль ковбоев" на Python
12. Разработка системы хранения истории болезней пациентов на Python
13. Разработка системы управления заказами в кафе на Python
14. Разработка простого текстового квеста на Python
15. Разработка игры "Угадай число" на Python
16. Разработка программы шифрования методом Цезаря на Python
17. Разработка генератора имен индейцев на Python
18. Разработка системы цензурирования слов в тексте на Python
19. Разработка системы управления турами туристического агенства на Python
20. Разработка модуля поиска совпадения слов и фраз в комментариях на Python
21. Разработка генератора имен предметов Diablo 3 на Python
22. Разработка игры "Скачки" на Python
23. Разработка функции хэширования строки на Python
24. Разработка игры "Бросок мяча в кольцо" на Python
25. Разработка системы управления автопарком такси на Python
26. Разработка русско-английского переводчика на Python
27. Разработка модуля музыкального плейлиста на Python

28. Разработка архиватора файлов в zip-архив на Python
29. Разработка системы управления списком покупок на Python
30. Разработка игры "Крестики-нолики" на Python
31. Разработка функции генератора случайных строк на Python
32. Разработка утилиты для массового удаления файлов на Python
33. Разработка парсера текста на Python

8 ДОПОЛНИТЕЛЬНОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Следует начать с определения, **Жизненный цикл программного обеспечения** (Software Life Cycle Model) — это период времени, который начинается с момента принятия решения о создании программного продукта и заканчивается в момент его полного изъятия из эксплуатации. Этот цикл — процесс построения и развития ПО.

Модели Жизненного цикла программного обеспечения

Жизненный цикл можно представить в виде моделей. В настоящее время наиболее распространенными являются: *каскадная*, *инкрементная* (*поэтапная модель с промежуточным контролем*) и *спиральная* модели жизненного цикла.

Каскадная модель

Каскадная модель (англ. *waterfall model*) — модель процесса разработки программного обеспечения, жизненный цикл которой выглядит как поток, последовательно проходящий фазы анализа требований, проектирования. реализации, тестирования, интеграции и поддержки.

Процесс разработки реализуется с помощью упорядоченной последовательности независимых шагов. Модель предусматривает, что каждый последующий шаг начинается после полного завершения выполнения предыдущего шага. На всех шагах модели выполняются вспомогательные и организационные процессы и работы, включающие управление проектом, оценку и управление качеством, верификацию и аттестацию, менеджмент конфигурации, разработку документации. В результате завершения шагов формируются промежуточные продукты, которые не могут изменяться на последующих шагах.

Жизненный цикл традиционно разделяют на следующие основные *этапы*:

1. Анализ требований,
2. Проектирование,
3. Кодирование (программирование),
4. Тестирование и отладка,
5. Эксплуатация и сопровождение.

Каскадная модель Жизненного цикла

Достоинства модели:

- стабильность требований в течение всего жизненного цикла разработки;
- на каждой стадии формируется законченный набор проектной документации, отвечающий критериям полноты и согласованности;
- определенность и понятность шагов модели и простота её применения;

- выполняемые в логической последовательности этапы работ позволяют планировать сроки завершения всех работ и соответствующие ресурсы (денежные. материальные и людские).

Каскадная модель хорошо зарекомендовала себя при построении относительно простых ПО, когда в самом начале разработки можно достаточно точно и полно сформулировать все требования к продукту.

Недостатки модели:

- сложность чёткого формулирования требований и невозможность их динамического изменения на протяжении пока идет полный жизненный цикл;
- низкая гибкость в управлении проектом;
- последовательность линейной структуры процесса разработки, в результате возврат к предыдущим шагам для решения возникающих проблем приводит к увеличению затрат и нарушению графика работ;
- непригодность промежуточного продукта для использования;
- невозможность гибкого моделирования уникальных систем;
- позднее обнаружение проблем, связанных со сборкой, в связи с одновременной интеграцией всех результатов в конце разработки;
- недостаточное участие пользователя в создании системы — в самом начале (при разработке требований) и в конце (во время приёмочных испытаний);
- пользователи не могут убедиться в качестве разрабатываемого продукта до окончания всего процесса разработки. Они не имеют возможности оценить качество, т.к.нельзя увидеть готовый продукт разработки;
- у пользователя нет возможности постепенно привыкнуть к системе. Процесс обучения происходит в конце жизненного цикла, когда ПО уже запущено в эксплуатацию;
- каждая фаза является предпосылкой для выполнения последующих действий, что превращает такой метод в рискованный выбор для систем, не имеющих аналогов, т.к. он не поддается гибкому моделированию.

Реализовать Каскадную модель жизненного цикла затруднительно ввиду сложности разработки ПС без возвратов к предыдущим шагам и изменения их результатов для устранения возникающих проблем.

Область применения Каскадной модели

Ограничение области применения каскадной модели определяется её недостатками. Её использование наиболее эффективно в следующих случаях:

1. при разработке проектов с четкими, неизменяемыми в течение *жизненного цикла* требованиями, понятными реализацией и техническими методиками;
2. при разработке проекта, ориентированного на построение системы или продукта такого же типа, как уже разрабатывались разработчиками ранее;
3. при разработке проекта, связанного с созданием и выпуском новой версии уже существующего продукта или системы;
4. при разработке проекта, связанного с переносом уже существующего продукта или системы на новую платформу;
5. при выполнении больших проектов, в которых задействовано несколько больших команд разработчиков.

Инкрементная модель

(поэтапная модель с промежуточным контролем)

Инкрементная модель (англ. *increment* — увеличение, приращение) подразумевает разработку программного обеспечения с линейной последовательностью стадий, но в несколько инкрементов (версий), т.е. с

запланированным улучшением продукта за все время пока Жизненный цикл разработки ПО не подойдет к окончанию.

Поэтапная модель с промежуточным контролем

Разработка программного обеспечения ведется итерациями с циклами обратной связи между этапами. Межэтапные корректировки позволяют учитывать реально существующее взаимовлияние результатов разработки на различных этапах, время жизни каждого из этапов растягивается на весь период разработки.

В начале работы над проектом определяются все основные требования к системе, подразделяются на более и менее важные. После чего выполняется разработка системы по принципу приращений, так, чтобы разработчик мог использовать данные, полученные в ходе разработки ПО. Каждый инкремент должен добавлять системе определенную функциональность. При этом выпуск начинают с компонентов с наивысшим приоритетом. Когда части системы определены, берут первую часть и начинают её детализировать, используя для этого наиболее подходящий процесс. В то же время можно уточнять требования и для других частей, которые в текущей совокупности требований данной работы были заморожены. Если есть необходимость, можно вернуться позже к этой части. Если часть готова, она поставляется клиенту, который может использовать её в работе. Это позволит клиенту уточнить требования для следующих компонентов. Затем занимаются разработкой следующей части системы. Ключевые этапы этого процесса — простая реализация подмножества требований к программе и совершенствование модели в серии последовательных релизов до тех пор, пока не будет реализовано ПО во всей полноте.

Жизненный цикл данной модели характерен при разработке сложных и комплексных систем, для которых имеется четкое видение (как со стороны заказчика, так и со стороны разработчика) того, что собой должен представлять конечный результат. Разработка версиями ведется в силу разного рода причин:

- отсутствия у заказчика возможности сразу профинансировать весь дорогостоящий проект;
- отсутствия у разработчика необходимых ресурсов для реализации сложного проекта в сжатые сроки;
- требований поэтапного внедрения и освоения продукта конечными пользователями. Внедрение всей системы сразу может вызвать у её пользователей неприятие и только “затормозить” процесс перехода на новые технологии. Образно говоря, они могут просто “не переварить большой кусок, поэтому его надо измельчить и давать по частям”.

Достоинства и **недостатки** этой модели (стратегии) такие же, как и у каскадной (классической модели жизненного цикла). Но в отличие от классической стратегии заказчик может раньше увидеть результаты. Уже по результатам разработки и внедрения первой версии он может незначительно изменить требования к

разработке, отказаться от нее или предложить разработку более совершенного продукта с заключением нового договора.

Достоинства:

- затраты, которые получаются в связи с изменением требований пользователей, уменьшаются, повторный анализ и совокупность документации значительно сокращаются по сравнению с каскадной моделью;
- легче получить отзывы от клиента о проделанной работе — клиенты могут озвучить свои комментарии в отношении готовых частей и могут видеть, что уже сделано. Т.к. первые части системы являются прототипом системы в целом.
- у клиента есть возможность быстро получить и освоить программное обеспечение — клиенты могут получить реальные преимущества от системы раньше, чем это было бы возможно с каскадной моделью.

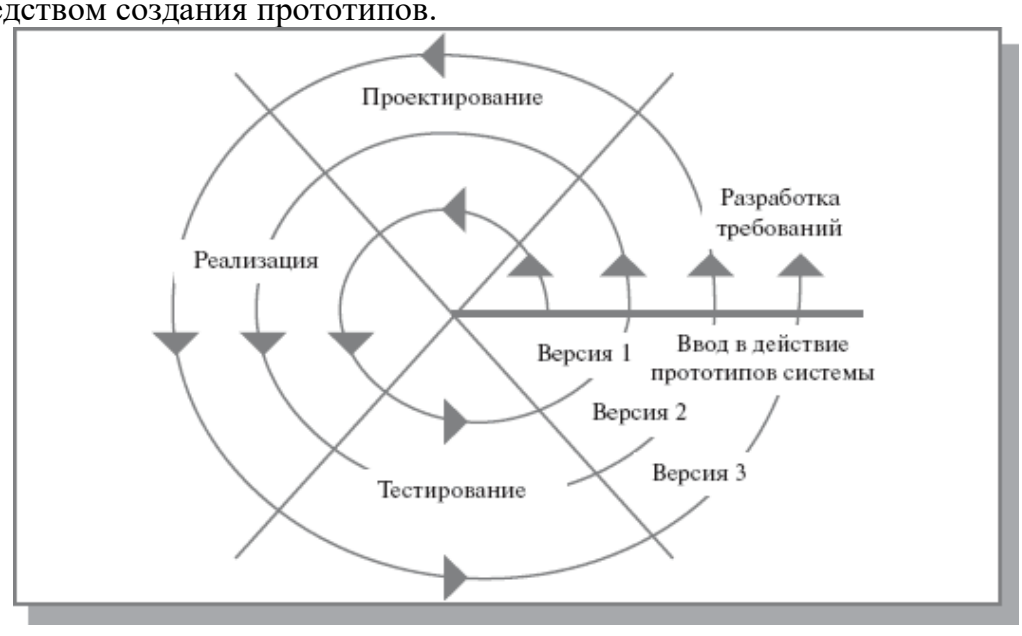
Недостатки модели:

- менеджеры должны постоянно измерять прогресс процесса. в случае быстрой разработки не стоит создавать документы для каждого минимального изменения версии;
- структура системы имеет тенденцию к ухудшению при добавлении новых компонентов — постоянные изменения нарушают структуру системы. Чтобы избежать этого требуется дополнительное время и деньги на рефакторинг. Плохая структура делает программное обеспечение сложным и дорогостоящим для последующих изменений. А прерванный Жизненный цикл ПО приводит еще к большим потерям.

Схема не позволяет оперативно учитывать возникающие изменения и уточнения требований к ПО. Согласование результатов разработки с пользователями производится только в точках, планируемых после завершения каждого этапа работ, а общие требования к ПО зафиксированы в виде технического задания на всё время её создания. Таким образом, пользователи зачастую получаю ПП, не удовлетворяющий их реальным потребностям.

Спиральная модель

Спиральная модель: Жизненный цикл — на каждом витке спирали выполняется создание очередной версии продукта, уточняются требования проекта, определяется его качество и планируются работы следующего витка. Особое внимание уделяется начальным этапам разработки — анализу и проектированию, где реализуемость тех или иных технических решений проверяется и обосновывается посредством создания прототипов.



Спиральная модель жизненного цикла

Данная модель представляет собой процесс разработки программного обеспечения, сочетающий в себе как проектирование, так и постадийное прототипировнаие с целью сочетания преимуществ восходящей и нисходящей концепции, делающая упор на начальные этапы жизненного цикла: анализ и проектирование. *Отличительной особенностью* этой модели является специальное внимание рискам, влияющим на организацию жизненного цикла.

На этапах анализа и проектирования реализуемость технических решений и степень удовлетворения потребностей заказчика проверяется путем создания прототипов. Каждый виток спирали соответствует созданию работоспособного фрагмента или версии системы. Это позволяет уточнить требования, цели и характеристики проекта, определить качество разработки, спланировать работы следующего витка спирали. Таким образом углубляются и последовательно конкретизируются детали проекта и в результате выбирается обоснованный вариант, который удовлетворяет действительным требованиям заказчика и доводится до реализации.

Жизненный цикл на каждом витке спирали — могут применяться разные модели процесса разработки ПО. В конечном итоге на выходе получается готовый продукт. Модель сочетает в себе возможности модели прототипирования и водопадной модели. Разработка итерациями отражает объективно существующий спиральный цикл создания системы. Неполное завершение работ на каждом этапе позволяет переходить на следующий этап, не дожидаясь полного завершения работы на текущем. Главная задача — как можно быстрее показать пользователям системы работоспособный продукт, тем самым активизируя процесс уточнения и дополнения требований.

Достоинства модели:

- позволяет быстрее показать пользователям системы работоспособный продукт, тем самым, активизируя процесс уточнения и дополнения требований;
- допускает изменение требований при разработке программного обеспечения, что характерно для большинства разработок, в том числе и типовых;
- в модели предусмотрена возможность гибкого проектирования, поскольку в ней воплощены преимущества каскадной модели, и в то же время разрешены итерации по всем фазам этой же модели;
- позволяет получить более надежную и устойчивую систему. По мере развития программного обеспечения ошибки и слабые места обнаруживаются и исправляются на каждой итерации;
- эта модель разрешает пользователям активно принимать участие при планировании, анализе рисков, разработке, а также при выполнении оценочных действий;
- уменьшаются риски заказчика. Заказчик может с минимальными для себя финансовыми потерями завершить развитие неперспективного проекта;
- обратная связь по направлению от пользователей к разработчикам выполняется с высокой частотой и на ранних этапах модели, что обеспечивает создание нужного продукта высокого качества.

Недостатки модели:

- если проект имеет низкую степень риска или небольшие размеры, модель может оказаться дорогостоящей. Оценка рисков после прохождения каждой спирали связана с большими затратами;
- Жизненный цикл модели имеет усложненную структуру, поэтому может быть затруднено её применение разработчиками, менеджерами и заказчиками;

- спираль может продолжаться до бесконечности, поскольку каждая ответная реакция заказчика на созданную версию может порождать новый цикл, что отдаляет окончание работы над проектом;
- большое количество промежуточных циклов может привести к необходимости в обработке дополнительной документации;
- использование модели может оказаться дорогостоящим и даже недопустимым по средствам, т.к. время. затраченное на планирование, повторное определение целей, выполнение анализа рисков и прототипирование, может быть чрезмерным;
- могут возникнуть затруднения при определении целей и стадий, указывающих на готовность продолжать процесс разработки на следующей и

Основная проблема спирального цикла — определение момента перехода на следующий этап. Для её решения вводятся временные ограничения на каждый из этапов *жизненного цикла* и переход осуществляется в соответствии с планом, даже если не вся запланированная работа закончена. *Планирование* производится на основе статистических данных, полученных в предыдущих проектах и личного опыта разработчиков.

Область применения спиральной модели

Применение спиральной модели целесообразно в следующих случаях:

- при разработке проектов, использующих новые технологии;
- при разработке новой серии продуктов или систем;
- при разработке проектов с ожидаемыми существенными изменениями или дополнениями требований;
- для выполнения долгосрочных проектов;
- при разработке проектов, требующих демонстрации качества и версий системы или продукта через короткий период времени;
- при разработке проектов. для которых необходим подсчет затрат, связанных с оценкой и разрешением рисков.

РЕЦЕНЗИЯ

на рабочую программу учебной дисциплины
МДК.01.01 Разработка программных модулей
для специальности 09.02.07 Информационные системы и программирование

Рабочая программа учебной дисциплины МДК.01.01 «Разработка программных модулей» соответствует ФГОС специальности среднего профессионального образования 09.02.07 Информационные системы и программирование, утвержденного приказом Министерства образования и науки Российской Федерации от «09» декабря 2016 г. № 1547, зарегистрирован в Министерстве юстиции России 26.12.2016 г. рег. № 44936 и примерной основной образовательной программе по специальности 09.02.07 Информационные системы и программирование (утвержденная протоколом Федерального учебно-методического объединения по УГПС 09.00.00 от 15 июля 2021 г. №3).

В рабочую программу учебной дисциплины включены разделы «Паспорт рабочей программы учебной дисциплины», «Структура и содержание учебной дисциплины», «Образовательные технологии», «Условия реализации программы учебной дисциплины», «Перечень основных и дополнительных информационных источников, необходимых для освоения дисциплины», «Методические рекомендации обучающимся по освоению дисциплины», «Оценочные средства для контроля успеваемости» и «Дополнительное обеспечение дисциплины».

Структура и содержание рабочей программы соответствуют целям образовательной программы СПО по специальности 09.02.07 «Информационные системы и программирование» и будущей профессиональной деятельности студента.

Объем рабочей программы учебной дисциплины полностью соответствует учебному плану подготовки по данной специальности. В программе четко сформулированы цели обучения, а также прогнозируемые результаты обучения по дисциплине.

На основании проведенной экспертиза можно сделать заключение, что рабочая программа учебной дисциплины МДК.01.01 «Разработка программных модулей» по специальности 09.02.07 «Информационные системы и программирование» соответствует требованиям стандарта, профессиональным требованиям, а также современным требованиям рынка труда.

Технический директор
ООО «ТехноСтарт» И.Г. Колодезный

« » 20 г.

РЕЦЕНЗИЯ

на рабочую программу учебной дисциплины
МДК.01.01 Разработка программных модулей
для специальности 09.02.07 Информационные системы и программирование

Рабочая программа учебной дисциплины МДК.01.01 «Разработка программных модулей» соответствует ФГОС специальности среднего профессионального образования 09.02.07 Информационные системы и программирование, утвержденного приказом Министерства образования и науки Российской Федерации от «09» декабря 2016 г. № 1547, зарегистрирован в Министерстве юстиции России 26.12.2016 г. рег. № 44936 и примерной основной образовательной программе по специальности 09.02.07 Информационные системы и программирование (утвержденная протоколом Федерального учебно-методического объединения по УГПС 09.00.00 от 15 июля 2021 г. №3).

В рабочую программу учебной дисциплины включены разделы «Паспорт рабочей программы учебной дисциплины», «Структура и содержание учебной дисциплины», «Образовательные технологии», «Условия реализации программы учебной дисциплины», «Перечень основных и дополнительных информационных источников, необходимых для освоения дисциплины», «Методические рекомендации обучающимся по освоению дисциплины», «Оценочные средства для контроля успеваемости» и «Дополнительное обеспечение дисциплины».

Структура и содержание рабочей программы соответствуют целям образовательной программы СПО по специальности 09.02.07 «Информационные системы и программирование» и будущей профессиональной деятельности студента.

Объем рабочей программы учебной дисциплины полностью соответствует учебному плану подготовки по данной специальности. В программе четко сформулированы цели обучения, а также прогнозируемые результаты обучения по дисциплине.

На основании проведенной экспертиза можно сделать заключение, что рабочая программа учебной дисциплины МДК.01.01 «Разработка программных модулей» по специальности 09.02.07 «Информационные системы и программирование» соответствует требованиям стандарта, профессиональным требованиям, а также современным требованиям рынка труда.

Технический директор ООО «ПРАЙ»
« » 20 г. Б.А. Шишкин