



1920

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.02.01 ТЕХНОЛОГИЯ РАЗРАБОТКИ ПРОГРАММНОГО
ОБЕСПЕЧЕНИЯ**

специальность 09.02.07 Информационные системы и программирование

Краснодар 2024

Рабочая программа учебной дисциплины МДК.02.01 ТЕХНОЛОГИЯ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ 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.02.01 ТЕХНОЛОГИЯ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ |
| Форма обучения | очная |
| Учебный год | 2024-2025 |
| 2 курс | 4 семестр |
| всего 103 часов, в том числе: | |
| лекции | 46 ч. |
| практические занятия | 48 ч. |
| курсовое проектирование | — |
| самостоятельные занятия | — |
| консультация | — |
| промежуточная аттестация | 9 ч. |
| форма итогового контроля | экзамен |

Составитель: преподаватель  А.Н. Чернышев

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.02.01 «Технология разработки программного обеспечения»

Специальность среднего профессионального образования:
09.02.07 Информационные системы и программирование

СОГЛАСОВАНО:

Нач. УМО филиала



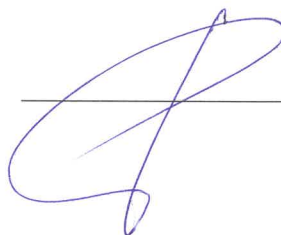
А.С. Демченко
«31» мая 2024 г.

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 | 5 |
| 1.1 | Область применения программы | 5 |
| 1.2 | Место учебной дисциплины в структуре программы подготовки специалистов среднего звена | 5 |
| 1.3 | Цели и задачи учебной дисциплины. Требования к результатам освоения учебной дисциплины..... | 5 |
| 1.4 | Перечень планируемых результатов обучения по дисциплине (Перечень формируемых компетенций)..... | 5 |
| 2 | СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ | 8 |
| 2.1 | Объем учебной дисциплины и виды учебной работы | 8 |
| 2.2 | Структура дисциплины..... | 8 |
| 2.3 | Тематический план и содержание учебной дисциплины МДК.02.01 ТЕХНОЛОГИЯ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ | 8 |
| 2.4 | Содержание разделов дисциплины | 11 |
| 2.4.1 | Занятия лекционного типа | 11 |
| 2.4.2 | Занятия семинарского типа | 11 |
| 2.4.3 | Практические занятия (Лабораторные занятия) | 11 |
| 2.4.4 | Содержание самостоятельной работы (Примерная тематика рефератов) | 12 |
| 2.4.5 | Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине..... | 12 |
| 3 | ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ | 14 |
| 3.1 | Образовательные технологии при проведении лекций | 14 |
| 3.2 | Образовательные технологии при проведении практических занятий | 14 |
| 4 | УСЛОВИЯ РЕАЛИЗАЦИИ УЧЕБНОЙ ДИСЦИПЛИНЫ | 15 |
| 4.1 | Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине | 15 |
| 4.2 | Перечень необходимого программного обеспечения..... | 15 |
| 5 | ПЕРЕЧЕНЬ ОСНОВНОЙ И ДОПОЛНИТЕЛЬНОЙ УЧЕБНОЙ ЛИТЕРАТУРЫ, НЕОБХОДИМОЙ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ | 16 |
| 5.1 | Основная литература | 16 |
| 5.2 | Дополнительная литература | 16 |
| 5.3 | Периодические издания | 16 |
| 5.4 | Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины..... | 17 |
| 6 | МЕТОДИЧЕСКИЕ УКАЗАНИЯ ОБУЧАЮЩИМСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ | 19 |
| 7 | КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ | 20 |
| 7.1 | Паспорт фонда оценочных средств | 20 |
| 7.2 | Критерии оценки знаний..... | 20 |
| 7.3 | Оценочные средства для проведения текущей аттестации | 22 |
| 7.4 | Оценочные средства для проведения промежуточной аттестации | 26 |
| 7.4.1 | Примерные вопросы для проведения промежуточной аттестации | 26 |
| 7.4.2 | Примерные задачи для проведения промежуточной аттестации..... | 27 |
| 8 | ДОПОЛНИТЕЛЬНОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ | 28 |

1 ОБЩАЯ ХАРАКТЕРИСТИКА РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ МДК.02.01 ТЕХНОЛОГИЯ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

1.1 Область применения программы

Рабочая программа учебной дисциплины МДК.02.01 «Технология разработки программного обеспечения» 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 «Технология разработки программного обеспечения» 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:

- модели процесса разработки программного обеспечения;
- основные принципы процесса разработки программного обеспечения;
- основные подходы к интегрированию программных модулей;
- основы верификации и аттестации программного обеспечения

В результате освоения дисциплины обучающийся должен уметь:

- использовать выбранную систему контроля версий;
- использовать методы для получения кода с заданной функциональностью и степенью качества

В результате освоения дисциплины обучающийся должен *знать*:

- модели процесса разработки программного обеспечения;
- основные принципы процесса разработки программного обеспечения;
- основные подходы к интегрированию программных модулей;
- основы верификации и аттестации программного обеспечения

Максимальная учебная нагрузка обучающегося 103 часов, в том числе:

- обязательная аудиторная учебная нагрузка обучающегося 94 часа;
- промежуточная аттестация 9 часов.

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

Освоение дисциплины «Технология разработки программного обеспечения» способствует формированию у студентов следующих профессиональных компетенций:

ПК 2.1. Разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент.

ПК 2.4 Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.

ПК 2.5 Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования.

| № п.п. | Индекс компетенции | Содержание компетенции (или её части) | В результате изучения учебной дисциплины обучающиеся должны | | |
|--------|--------------------|--|--|---|---|
| | | | знать | уметь | иметь практический опыт |
| 1 | ПК 2.1. | Разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. | <p>Модели процесса разработки программного обеспечения. Основные принципы процесса разработки программного обеспечения. Основные подходы к интегрированию программных модулей. Виды и варианты интеграционных решений. Современные технологии и инструменты интеграции. Основные протоколы доступа к данным. Методы и способы идентификации сбоев и ошибок при интеграции приложений. Методы отладочных классов.</p> | <p>Анализировать проектную и техническую документацию. Использовать специализированные графические средства построения и анализа архитектуры программных продуктов. Организовывать заданную интеграцию модулей в программные средства на базе имеющейся архитектуры и автоматизации бизнес-процессов. Определять источники и приемники данных.</p> | <p>Разрабатывать и оформлять требования к программным модулям по предложенной документации. Разрабатывать тестовые наборы (пакеты) для программного модуля.</p> |
| 2 | ПК 2.4 | Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения. | <p>Модели процесса разработки программного обеспечения. Основные принципы процесса разработки программного обеспечения. Основные подходы к интегрированию программных модулей. Основы верификации и аттестации программного обеспечения. Методы и способы идентификации сбоев и ошибок при интеграции приложений. Методы и схемы обработки исключительных ситуаций. Основные методы и виды тестирования программных продуктов.</p> | <p>Использовать выбранную систему контроля версий. Анализировать проектную и техническую документацию. Выполнять тестирование интеграции. Организовывать постобработку данных. Использовать приемы работы в системах контроля версий. Оценивать размер минимального набора тестов. Разрабатывать тестовые пакеты и тестовые сценарии. Выполнять ручное и автоматизированное тестирование программного модуля. Выявлять ошибки в системных компонентах на основе спецификаций.</p> | <p>Разрабатывать тестовые наборы (пакеты) для программного модуля. Разрабатывать тестовые сценарии программного средства. Инспектировать разработанные программные модули на предмет соответствия стандартам кодирования.</p> |

| | | | | | |
|---|--------|---|---|--|--|
| 3 | ПК 2.5 | <p>Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования.</p> | <p>Модели процесса разработки программного обеспечения. Основные принципы процесса разработки программного обеспечения. Основные подходы к интегрированию программных модулей. Основы верификации и аттестации программного обеспечения. Стандарты качества программной документации. Основы организации инспектирования и верификации. Встроенные и основные специализированные инструменты анализа качества программных продуктов. Методы организации работы в команде разработчиков.</p> | <p>Использовать выбранную систему контроля версий. Использовать методы для получения кода с заданной функциональностью и степенью качества. Анализировать проектную и техническую документацию. Организовывать постобработку данных. Приемы работы в системах контроля версий. Выявлять ошибки в системных компонентах на основе спецификаций.</p> | <p>Инспектировать разработанные программные модули на предмет соответствия стандартам кодирования.</p> |
|---|--------|---|---|--|--|

2 СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

2.1. Объем учебной дисциплины и виды учебной работы

| Вид учебной работы | Всего часов | Семестр |
|------------------------------------|-------------|---------|
| | | 4 |
| Учебная нагрузка (всего) | 103 | 103 |
| Аудиторная нагрузка (всего) | 94 | 94 |
| в том числе: | | |
| лекционные занятия | 46 | 46 |
| практические занятия | 50 | 50 |
| Курсовое проектирование | – | – |
| Самостоятельная работа | – | – |
| Консультации | – | – |
| Промежуточная аттестация – экзамен | 9 | 9 |

2.2 Структура дисциплины

Освоение учебной дисциплины МДК.02.01 ТЕХНОЛОГИЯ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ включает изучение следующих разделов и тем:

| № | Тема | Всего часов | Лекции | Практические | Самостоятельная работа |
|---|---|-------------|--------|--------------|------------------------|
| | | 82 | 46 | 34 | |
| 1 | <i>Тема 2.1.1 Основные понятия и стандартизация требований к программному обеспечению</i> | 32 | 18 | 14 | |
| 2 | <i>Тема 2.1.2. Описание и анализ требований. Диаграммы IDEF</i> | 32 | 14 | 18 | |
| 3 | <i>Тема 2.1.3. Оценка качества программных средств</i> | 32 | 14 | 18 | |

2.3 Тематический план и содержание учебной дисциплины

МДК.02.01 ТЕХНОЛОГИЯ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

| Наименование разделов и тем профессионального модуля (ПМ), междисциплинарных | Содержание учебного материала, лабораторные работы и практические занятия, внеаудиторная (самостоятельная) учебная работа обучающихся, курсовая работа (проект) | Объем в часах |
|---|---|---------------|
| Тема 2.1.1 Основные понятия и стандартизация требований к программному обеспечению | <p>Содержание</p> <p>1. Понятия требований, классификация, уровни требований. Методологии и стандарты, регламентирующие работу с требованиями.</p> <p>2. Современные принципы и методы разработки программных приложений.</p> <p>3. Методы организации работы в команде разработчиков. Системы контроля версий</p> <p>4. Основные подходы к интегрированию программных модулей.</p> <p>5. Стандарты кодирования.</p> | 32 |

| | | |
|--|---|----|
| | <p>В том числе практических занятий и лабораторных работ</p> <p>1. Практическое занятие «Анализ предметной области»</p> <p>2. Практическое занятие «Разработка и оформление технического задания»</p> <p>3. Практическое занятие «Построение архитектуры программного средства»</p> <p>4. Практическое занятие «Изучение работы в системе контроля версий»</p> | 14 |
| <p>Тема 2.1.2. Описание и анализ требований. Диаграммы IDEF</p> | <p>Содержание</p> <p>Описание требований: унифицированный язык моделирования - краткий словарь. Диаграммы UML.</p> <p>1. Описание и оформление требований (спецификация). Анализ требований и стратегии выбора решения</p> | 32 |
| | <p>В том числе практических занятий и лабораторных работ</p> <p>1. Лабораторная работа «Построение Диаграммы Вариантов использования и Диаграммы. Последовательности»</p> <p>2. Лабораторная работа «Построение Диаграммы Кооперации и Диаграммы Развертывания»</p> <p>3. Лабораторная работа «Построение Диаграммы Деятельности, Диаграммы Состояний и Диаграммы Классов»</p> <p>4. Лабораторная работа «Построение Диаграммы компонентов»</p> <p>5. Лабораторная работа «Построение Диаграмм потоков Данных»</p> | 18 |
| <p>Тема 2.1.3. Оценка качества программных средств</p> | <p>Содержание</p> <p>1. Цели и задачи и виды тестирования. Стандарты качества программной документации. Меры и метрики.</p> <p>2. Тестовое покрытие.</p> <p>3. Тестовый сценарий, тестовый пакет.</p> <p>4. Анализ спецификаций. Верификация и аттестация программного обеспечения.</p> | 32 |
| | <p>В том числе практических занятий и лабораторных работ</p> <p>1. Лабораторная работа «Разработка тестового сценария»</p> <p>2. Лабораторная работа «Оценка необходимого количества тестов»</p> <p>3. Лабораторные работы «Разработка тестовых пакетов»</p> <p>4. Лабораторные работы «Оценка программных средств с помощью метрик»</p> | 18 |

| | | |
|--|---|------------|
| | 5. Лабораторные работы «Инспекция программного кода на предмет соответствия стандартам кодирования» | |
| | Итоговая аттестация | 9 |
| | Всего | 103 |

2.4 Содержание разделов дисциплины

2.4.1 Занятия лекционного типа

| № п/п | Наименование раздела | Содержание раздела | Форма текущего контроля |
|-------|---|---|-------------------------|
| 1 | <i>Тема 2.1.1 Основные понятия и стандартизация требований к программному обеспечению</i> | <ol style="list-style-type: none"> 1. Понятия требований, классификация, уровни требований. Методологии и стандарты, регламентирующие работу с требованиями. 2. Современные принципы и методы разработки программных приложений. 3. Методы организации работы в команде разработчиков. Системы контроля версий 4. Основные подходы к интегрированию программных модулей. 5. Стандарты кодирования. | У, КР |
| 2 | <i>Тема 2.1.2. Описание и анализ требований. Диаграммы IDEF</i> | <p>Описание требований: унифицированный язык моделирования - краткий словарь. Диаграммы UML.</p> <ol style="list-style-type: none"> 1. Описание и оформление требований (спецификация). Анализ требований и стратегии выбора решения | У, КР |
| 3 | <i>Тема 2.1.3. Оценка качества программных средств</i> | <ol style="list-style-type: none"> 1. Цели и задачи и виды тестирования. Стандарты качества программной документации. Меры и метрики. 2. Тестовое покрытие. 3. Тестовый сценарий, тестовый пакет. 4. Анализ спецификаций. Верификация и аттестация программного обеспечения. | 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 | <i>Тема 2.1.1 Основные понятия и стандартизация требований к программному обеспечению</i> | <ol style="list-style-type: none"> 1. Практическое занятие «Анализ предметной области» 2. Практическое занятие «Разработка и оформление технического задания» 3. Практическое занятие «Построение архитектуры программного средства» 4. Практическое занятие «Изучение работы в системе контроля версий» | ПР |

| | | | |
|---|--|---|-----------|
| 2 | <p>Тема 2.1.2. Описание и анализ требований. Диаграммы IDEF</p> | <ol style="list-style-type: none"> 1. Лабораторная работа «Построение Диаграммы Вариантов использования и Диаграммы Последовательности» 2. Лабораторная работа «Построение Диаграммы Кооперации и Диаграммы Развертывания» 3. Лабораторная работа «Построение Диаграммы Деятельности, Диаграммы Состояний и Диаграммы Классов» 4. Лабораторная работа «Построение Диаграммы компонентов» 5. Лабораторная работа «Построение Диаграмм потоков Данных» | <p>ПР</p> |
| 3 | <p>Тема 2.1.3. Оценка качества программных средств</p> | <ol style="list-style-type: none"> 1. Лабораторная работа «Разработка тестового сценария» 2. Лабораторная работа «Оценка необходимого количества тестов» 3. Лабораторные работы «Разработка тестовых пакетов» 4. Лабораторные работы «Оценка программных средств с помощью метрик» 5. Лабораторные работы «Инспекция программного кода на предмет соответствия стандартам кодирования» | <p>ПР</p> |

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 «Технология разработки программного обеспечения» 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 |
|---|---|----------|
| <i>Тема 2.1.1 Основные понятия и стандартизация требований к программному обеспечению</i> | Аудиовизуальная технология, технология развивающего обучения | 18* |
| <i>Тема 2.1.2. Описание и анализ требований. Диаграммы IDEF</i> | Аудиовизуальная технология, личностно-деятельностное обучение | 14* |
| <i>Тема 2.1.3. Оценка качества программных средств</i> | Аудиовизуальная технология, личностно-деятельностное обучение | 14* |
| Итого | | 46 |
| В том числе интерактивное | | 46 |

3.2. Образовательные технологии при проведении практических занятий

| Тема | Виды применяемых образовательных технологий | Кол. час |
|---|--|----------|
| <i>Тема 2.1.1 Основные понятия и стандартизация требований к программному обеспечению</i> | Технология личностно-деятельностного обучения, технология проблемного обучения | 14* |
| <i>Тема 2.1.2. Описание и анализ требований. Диаграммы IDEF</i> | Технология личностно-деятельностного обучения, технология проблемного обучения | 18* |
| <i>Тема 2.1.3. Оценка качества программных средств</i> | Технология проблемного обучения | 18* |
| Итого | | 34 |
| В том числе интерактивное | | 34 |

4 УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ МДК.02.01 ТЕХНОЛОГИЯ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

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

Лаборатории *Программного обеспечения и сопровождения компьютерных систем*, оснащенные в соответствии с п. 6.1.2.1. Примерной программы по *специальности*

Оснащенные базы практики, в соответствии с п 6.1.2.3 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 Reader (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. Гагарина, Л. Г. Технология разработки программного обеспечения : учебное пособие / Л.Г. Гагарина, Е.В. Кокорева, Б.Д. Сидорова-Виснадул ; под ред. Л.Г. Гагариной. — Москва : ФОРУМ : ИНФРА-М, 2022. — 400 с. — (Среднее профессиональное образование). - ISBN 978-5-8199-0812-9. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1794453>. – Режим доступа: по подписке.

2. Черткова, Е. А. Программная инженерия. Визуальное моделирование программных систем : учебник для среднего профессионального образования / Е. А. Черткова. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2021. — 147 с. — (Профессиональное образование). — ISBN 978-5-534-09823-5. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/473307>

5.2 Дополнительная литература

1. Гниденко, И. Г. Технология разработки программного обеспечения : учебное пособие для среднего профессионального образования / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — Москва : Издательство Юрайт, 2021. — 235 с. — (Профессиональное образование). — ISBN 978-5-534-05047-9. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/472502>.

2. Гагарина, Л. Г. Введение в архитектуру программного обеспечения : учебное пособие / Л.Г. Гагарина, А.Р. Федоров, П.А. Федоров. — Москва : ФОРУМ : ИНФРА-М, 2021. — 320 с. — (Среднее профессиональное образование). - ISBN 978-5-8199-0903-4. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1236297>. – 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 издания, 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 «Znaniy.com» [учебные, научные, научно-популярные материалы различных издательств, журналы] : сайт. – URL: <http://znaniy.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.systems.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 «Технология разработки программного обеспечения» 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 «Технология разработки программного обеспечения»; 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.02.01 ТЕХНОЛОГИЯ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ»

7.1 Паспорт фонда оценочных средств

| № п/п | Контролируемые разделы (темы) дисциплины | Компетенции | Наименование оценочного средства |
|-------|---|------------------------|------------------------------------|
| 1. | <i>Тема 2.1.1 Основные понятия и стандартизация требований к программному обеспечению</i> | ПК 2.1, ПК 2.4, ПК 2.5 | Проверка конспектов, устный опрос, |
| 2. | <i>Тема 2.1.2. Описание и анализ требований. Диаграммы IDEF</i> | ПК 2.1, ПК 2.4, ПК 2.5 | Проверка конспектов, тест |
| 3. | <i>Тема 2.1.3. Оценка качества программных средств</i> | ПК 2.1, ПК 2.4, ПК 2.5 | Проверка конспектов, 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 1. Разработка программного обеспечения | | |
| ПК 2.1 Разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент | <p>Оценка «отлично» - разработан и обоснован вариант интеграционного решения с помощью графических средств среды разработки, указано хотя бы одно альтернативное решение; бизнес-процессы учтены в полном объеме; вариант оформлен в полном соответствии с требованиями стандартов; результаты верно сохранены в системе контроля версий.</p> <p>Оценка «хорошо» - разработана и прокомментирована архитектура варианта интеграционного решения с помощью графических средств, учтены основные бизнес-процессы; вариант оформлен в соответствии с требованиями стандартов; результаты сохранены в системе контроля версий.</p> <p>Оценка «удовлетворительно» - разработана и архитектура варианта интеграционного решения с помощью графических средств, учтены основные бизнес-процессы с незначительными упущениями; вариант оформлен в соответствии с требованиями стандартов с некоторыми отклонениями; результат сохранен в системе контроля версий.</p> | <p>Экзамен/зачет в форме собеседования:</p> <p>- практическое задание по формированию требований к программным модулям в соответствии с техническим заданием. Защита отчетов по практическим и лабораторным работам. Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики</p> |

| | | |
|---|---|--|
| <p>ПК 2.4 Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения</p> | <p>Оценка «отлично» - обоснован размер тестового покрытия, разработан тестовый сценарий и тестовые пакеты в соответствии с этим сценарием в соответствии с минимальным размером тестового покрытия, выполнено тестирование интеграции и ручное тестирование, выполнено тестирование с применением инструментальных средств, выявлены ошибки системных компонент (при наличии), заполнены протоколы тестирования.</p> <p>Оценка «хорошо»- обоснован размер тестового покрытия, разработан тестовый сценарий и тестовые пакеты в соответствии с этим сценарием, выполнено тестирование интеграции и ручное тестирование, выполнено тестирование с применением инструментальных средств, заполнены протоколы тестирования. Оценка «удовлетворительно»- определен размер тестового покрытия, разработан тестовый сценарий и тестовые пакеты, выполнено тестирование интеграции и ручное тестирование, частично выполнено тестирование с применением инструментальных средств, частично заполнены протоколы тестирования.</p> | <p>Экзамен/зачет в форме собеседования: практическое задание по разработке тестовых сценариев и наборов для заданных видов тестирования и выполнение тестирования.</p> <p>Защита отчетов по практическим и лабораторным работам Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики</p> |
| <p>ПК 2.5 Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования</p> | <p>Оценка «отлично» - продемонстрировано знание стандартов кодирования более чем одного языка программирования, выявлены все имеющиеся несоответствия стандартам в предложенном коде.</p> <p>Оценка «хорошо» - продемонстрировано знание стандартов кодирования более чем одного языка программирования, выявлены существенные имеющиеся несоответствия стандартам в предложенном коде.</p> <p>Оценка «удовлетворительно» - продемонстрировано знание стандартов кодирования языка программирования, выявлены некоторые несоответствия стандартам в предложенном коде.</p> | <p>Экзамен/зачет в форме собеседования: практическое задание по инспектированию программного кода Защита отчетов по практическим и лабораторным работам Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики</p> |

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. Понятия требований, классификация, уровни требований.
2. Методологии и стандарты, регламентирующие работу с требованиями.
3. Современные принципы и методы разработки программных приложений.
4. Методы организации работы в команде разработчиков.
5. Системы контроля версий
6. Основные подходы к интегрированию программных модулей.
7. Стандарты кодирования.
8. Описание требований: унифицированный язык моделирования - краткий словарь.
9. Диаграммы UML.
10. Описание и оформление требований (спецификация).
11. Анализ требований и стратегии выбора решения
12. Цели и задачи и виды тестирования.
13. Стандарты качества программной документации.

14. Меры и метрики.
15. Тестовое покрытие.
16. Тестовый сценарий
17. Тестовый пакет.

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

1. В каких единицах можно измерить надежность разработанного программного продукта:
 1. **отказов/час;**
 2. км/час;
 3. Кбайт/сек;
 4. операций/сек
2. В каких единицах можно измерить быстродействие ПО:
 1. отказов/час;
 2. км/час;
 3. Кбайт/сек;
 4. **операций/сек**
3. Что относится к этапу программирования:
 1. **написание кода программы;**
 2. разработка интерфейса;
 3. работоспособность;
 4. анализ требований.
4. Укажите правильную последовательность этапов программирования:
 1. **компилирование, компоновка, отладка;**
 2. компоновка, отладка, компилирование;
 3. отладка, компилирование, компоновка;
 4. компилирование, отладка, компоновка.
5. К инструментальным средствам программирования относятся:
 1. **компиляторы, интерпретаторы;**
 2. СУБД (системы управления базами данных);
 3. BIOS (базовая система ввода-вывода);
 4. ОС (операционные системы).
6. Доступ, при котором записи файла читаются в физической последовательности, называется:
 1. прямым;
 2. простым;
 3. **последовательным;**
 4. основным
7. Какого метода программирования не существует:
 1. **логического;**
 2. структурного;
 3. модульного;
 4. объектно-ориентированного
8. Какой этап выполняется раньше остальных:
 1. **разработка алгоритма;**
 2. выбор языка программирования;
 3. написание исходного кода;
 4. компиляция
9. Наличие комментариев позволяет:

1. быстрее писать программы;
 2. быстрее выполнять программы;
 3. **быстрее найти ошибки в программе;**
 4. быстрее произвести описание структуры программы
10. Что определяет выбор языка программирования:
1. **область приложения;**
 2. знание языка;
 3. наличие дополнительных библиотек;
 4. особенности структуры
11. Для каких задач характерно использование большого количества исходных данных, выполнение операций поиска, группировки:
1. **для экономических задач;**
 2. для системных задач;
 3. для инженерных задач;
 4. для математический
12. На каком этапе производится выбор языка программирования:
1. **проектирование;**
 2. программирование;
 3. отладка;
 4. тестирование.
13. Когда приступают к тестированию программы:
1. **когда программа уже закончена;**
 2. после постановки задачи;
 3. на этапе программирования;
 4. на этапе проектирования;
14. Одним из методов автоматизации программирования является:
1. структурное программирование;
 2. модульное программирование;
 3. **визуальное программирование;**
 4. объектно-ориентированное программирование.
15. Критерием оптимизации программы является:
1. **быстродействие или размер программы;**
 2. быстродействие и размер программы;
 3. надежность или эффективность;
 4. надежность и эффективность
16. Что из перечисленного относится к специфическим особенностям ПО как продукта:
1. **низкие затраты при дублировании;**
 2. универсальность;
 3. простота эксплуатации;
 4. наличие поддержки (сопровождения) со стороны разработчика
17. Этап, занимающий наибольшее время, в жизненном цикле программы:
1. **сопровождение;**
 2. проектирование;
 3. тестирование;
 4. программирование;
18. Этап, занимающий наибольшее время, при разработке программы:
1. **тестирование;**
 2. сопровождение;
 3. проектирование;
 4. программирование;

19. Первый этап в жизненном цикле программы:
 1. анализ требований;
 2. **формулирование требований;**
 3. проектирование;
 4. автономное тестирование;
20. Один из необязательных этапов жизненного цикла программы:
 1. проектирование;
 2. тестирование;
 3. программирование;
 4. **оптимизация**
21. Самый большой этап в жизненном цикле программы:
 1. **эксплуатация;**
 2. изучение предметной области;
 3. тестирование;
 4. корректировка ошибок
22. Какой этап выполняется раньше:
 1. тестирование;
 2. **отладка;**
 3. эксплуатация;
 4. оптимизация
23. Какой из этапов выполняется раньше остальных:
 1. отладка;
 2. оптимизация;
 3. **программирование;**
 4. тестирование
24. Что выполняется раньше:
 1. **компиляция;**
 2. отладка;
 3. компоновка;
 4. тестирование
25. В стадии разработки программы не входит:
 1. постановка задачи;
 2. составление спецификаций;
 3. **автоматизация программирования;**
 4. эскизный проект
26. Самый важный критерий качества программы:
 1. надежность;
 2. **работоспособность;**
 3. быстроедействие;
 4. простота эксплуатации
27. Один из способов оценки качества ПО:
 1. **сравнение с аналогами;**
 2. наличие документации;
 3. оптимизация программы;
 4. структурирование алгоритма
28. Существует ли связь между эффективностью и оптимизацией программы:
 1. **да;**
 2. нет;
 3. в случаях коллективной разработки ПО;
 4. в случаях индивидуальной разработки ПО

29. Наиболее важным критерием качества при разработке ПО является:

1. быстродействие;
2. удобство в эксплуатации;
3. **надежность;**
4. эффективность

30. Одним из способов оценки надежности ПО является:

1. сравнение с аналогами;
2. трассировка;
3. оптимизация;
4. **тестирование**

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. Понятия требований, классификация, уровни требований.
2. Методологии и стандарты, регламентирующие работу с требованиями.
3. Современные принципы и методы разработки программных приложений.
4. Методы организации работы в команде разработчиков.
5. Системы контроля версий
6. Основные подходы к интегрированию программных модулей.
7. Стандарты кодирования.
8. Описание требований: унифицированный язык моделирования - краткий словарь.
9. Диаграммы UML.
10. Описание и оформление требований (спецификация).
11. Анализ требований и стратегии выбора решения
12. Цели и задачи и виды тестирования.
13. Стандарты качества программной документации.
14. Меры и метрики.

15. Тестовое покрытие.
16. Тестовый сценарий
17. Тестовый пакет.

7.4.2 Примерные задачи для проведения промежуточной аттестации

1. Анализ выбранного стиля программирования
2. Разработка проекта программного обеспечения
3. Разработка структурного алгоритма
4. Разработка программного продукта с использованием объектно-ориентированного программирования
5. Разработка справочной системы
6. Тестирование методом «белого ящика»
7. Тестирование методом «черного ящика»
8. Способы анализа граничных решений
9. Способы диаграмм причин-следствий
10. Нисходящее тестирование интеграций
11. Восходящее тестирование интеграций
12. Анализ предметной области
13. Автоматизированное тестирование
14. Отладка программ
15. Оптимизация программ
16. Работа в составе бригады

8 ДОПОЛНИТЕЛЬНОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Современные принципы и средства разработки программного обеспечения

1. Современные методы разработки ПО

Метод нисходящего проектирования (метод пошаговой детализации, метод иерархического проектирования, top-down-подход)

Суть метода заключается в определении спецификаций компонентов системы путем последовательного выделения в ее составе отдельных составляющих и их постепенной детализации до уровня, обеспечивающего однозначное понимание того, что и как необходимо разрабатывать и реализовывать.

Этот метод является незаменимым при разработке сложных по характеру и больших по объему программ, когда к их разработке необходимо привлекать большое число программистов, работающих параллельно. Он позволяет концентрировать внимание разработчиков на наиболее ответственных частях программы, а также облегчает возможность постоянного контроля за ее работоспособностью по мере разработки, отладки и объединения отдельных составляющих программ за счет организации непрерывности этого процесса в течение всей разработки.

Для ускорения разработки программного комплекса часто вместо некоторых программ нижнего уровня, находящихся в процессе разработки, могут применяться специальные "программы-заглушки". Программы-заглушки требуются только на ранних стадиях разработки для того, чтобы не сдерживать общий ход создания программного комплекса. Суть программы-заглушки заключается в том, что при обращении к ней в соответствии с заданным набором исходных тестовых данных она не формирует, а выбирает результат "решения" из заранее подготовленного набора. Благодаря этому обеспечивается возможность имитировать работу на ЭВМ реально создаваемой программы, а следовательно, осуществлять проверку работоспособности программ верхнего уровня еще до того, как будут разработаны и отлажены все составляющие программы нижнего уровня.

Модульное проектирование

Реализация метода *нисходящего проектирования* тесно связана с другим понятием программирования - *модульным проектированием*, так как на практике при декомпозиции сложной программы возникает вопрос о разумном пределе ее дробления на составные части. Вместе с тем понятие модульности нельзя сводить только к представлению сложных программных комплексов в виде набора отдельных функциональных блоков.

Модуль - это последовательность логически взаимосвязанных фрагментов задачи, оформленных как отдельная часть программы. При этом программные модули должны обладать следующими свойствами:

- на модуль можно ссылаться (т.е. обращаться к нему) по имени, в том числе и из других модулей;
- по завершении работы модуль должен возвращать управление тому модулю, который его вызывал;
- модуль должен иметь один вход и выход;
- модуль должен иметь небольшой размер, обеспечивающий его обзорность.

При разработке сложных программ в них выделяют головной управляющий модуль, подчиненные ему модули, обеспечивающие реализацию отдельных функций управления, функциональную обработку (т.е. непосредственную реализацию основного назначения программного комплекса), а также вспомогательные модули, обеспечивающие сервисное обслуживание пакета (например, сбор и анализ статистики работы программы, обработка различного рода ошибочных ситуаций, обучение и выдача подсказок и т.п.).

Модульный принцип разработки программ обладает следующими преимуществами:

- большую программу могут разрабатывать одновременно несколько исполнителей, и это позволяет сократить сроки ее разработки;
- появляется возможность создавать и многократно использовать в дальнейшем библиотеки наиболее употребляемых программ;

- упрощается процедура загрузки больших программ в оперативную память, когда требуется ее сегментация;
- возникает много естественных контрольных точек для наблюдения за осуществлением хода разработки программ, а в последующем для контроля за ходом исполнения программ;
- обеспечивается более эффективное тестирование программ, проще осуществляются проектирование и последующая отладка.
- Преимущества модульного принципа построения программ особенно наглядно проявляются на этапе сопровождения и модификации программных продуктов, позволяя значительно сократить затраты сил и средств на реализацию этого этапа.

Структурное программирование

Актуальная для начального периода развития и использования ЭВМ проблема разработки программ, занимающих минимум основной памяти и выполняющихся за кратчайшее время, в последующем в связи резким падением стоимости аппаратной части ЭВМ, значительным возрастанием их быстродействия и объемов памяти сменилась необходимостью разработки и применения принципиально новых методов составления программ. Все это нашло свое воплощение в разработке принципа *структурного программирования*. Одной из целей структурного программирования было стремление облегчить разработку и отладку программных модулей, а главное - их последующее сопровождение и модификацию.

В настоящее время структурное программирование - это целая дисциплина, объединяющая несколько взаимосвязанных способов создания ясных, легких для понимания программ. Эффективность применения современных универсальных языков программирования во многом определяется удобством написания с их помощью структурных программ.

CASE-технологии

За последнее десятилетие в области средств автоматизации программирования сформировалось новое направление под общим названием [CASE-технологии](#) (Computer Aided Software Engineering).

CASE-технология представляет собой совокупность средств системного анализа, проектирования, разработки и сопровождения сложных программных систем, поддерживаемых комплексом взаимоувязанных инструментальных средств автоматизации всех этапов разработки программ. Благодаря структурным методам CASE-технология на стадиях анализа и проектирования обеспечивает разработчиков широкими возможностями для различного рода моделирования, а централизованное хранение всей необходимой для проектирования информации и контроль за целостностью данных гарантируют согласованность взаимодействия всех специалистов, занятых в разработке ПО.

Технологии RAD

В начале 80-х годов появилась методология, по которой разработка программы начиналась не после завершения процесса выработки окончательных требований к ней, а как только устанавливались требования на первый, "стартовый" (пилотный) вариант прикладной программы, позволяющий начать содержательную работу по ее реализации на компьютере.

Это дало пользователю возможность, получая уже с первых шагов конкретное представление о характере реализации задачи, уточнять ее постановку. Тем самым облегчался процесс экспериментального поиска нужного решения автоматизации задачи. Благодаря тесному взаимодействию разработчика с заказчиком (пользователем) на самом ответственном этапе создания прикладных программ между ними достигалось быстрое взаимопонимание цели поставленной задачи и возможности ее автоматизации в данных конкретных условиях. Это повышало скорость разработки программ и послужило основанием для названия такой технологии *RAD* (*Rapid Application Development* - быстрая разработка программ), которая получила широкое распространение.

Data Warehouse

Другое направление разработки прикладных программных средств, олицетворяющее собой современный подход к реализации широкого круга задач для принятия [управленческих решений](#), базируется на концепции создания специального хранилища данных (*Data Warehouse*).

Основное отличие концепции Data Warehouse от традиционного представления баз данных заключается в следующем:

- во-первых, в том, что актуализация данных в Data Warehouse означает не обновление элементов информации, а добавление новых элементов к уже имеющимся (что расширяет возможности проведения различного рода сравнительного анализа);
- во-вторых, в том, что наряду с информацией, непосредственно отражающей состояние системы управления, в Data Warehouse аккумулируются и метаданные.

Метаданные (данные о данных) облегчают возможность визуального представления содержимого Data Warehouse, позволяют, "перемещаясь" по хранилищу, быстро отбирать необходимые данные для последующей обработки.

Основные типы метаданных Data Warehouse отражают:

- структуру и содержимое хранилища;
- соответствие между исходными и выходными данными;
- объемные характеристики данных;
- критерии архивирования;
- отношения между данными;
- информацию по кодированию;
- интервал жизни данных и т.п.

Концепция *Data Warehouse* поддерживается RAD средствами разработки прикладного ПО.

Концепция Data Warehouse обеспечивает возможность разработки программных приложений для поддержки процессов принятия решений с использованием OLAP-систем.

Система OLAP (On-Line Analytical Process) предоставляет возможность разработки информационных систем, ориентированных на уна организацию многомерных баз данных и создание корпоративных сетей, а также обеспечивает поддержку Web-технологий в сетях Internet/Intranet

Успешное применение инструментальных средств OLAP-систем объясняется быстротой разработки приложений, гибкостью и широкими возможностями в области доступа к данным и их преобразования. В настоящее время на рынке ПО предлагается большое число OLAP-стем, разработчиками которых являются различные фирмы, например IBM, Informix, Microsoft, Oracle, Sybase и др.

2. Инструментарий технологии программирования

Инструментарий технологии программирования - программные продукты поддержки (обеспечения) технологии программирования.

В рамках этого направления сформировались следующие группы программных продуктов:

1. средства для создания приложений, включающие:
 - локальные средства, обеспечивающие выполнение отдельных работ по созданию программ;
 - интегрированные среды разработчиков программ, обеспечивающие выполнение комплекса взаимосвязанных работ по созданию программ;
2. средства для создания информационных систем (CASE- технология), представляющие методы анализа, проектирования и создания программных систем и предназначенные для автоматизации процессов разработки и реализации информационных систем.

3. Средства для создания приложений

Локальные средства разработки программ

Эти средства на рынке программных продуктов наиболее представительны и включают языки и системы программирования, а также инструментальную среду пользователя.

Язык программирования - формализованный язык для описания алгоритма решения задачи на компьютере.

Средства для создания приложений - совокупность языков и систем программирования, а также различные программные комплексы для отладки и поддержки создаваемых программ.

Языки программирования можно условно разделить на следующие классы (если в качестве признака классификации взять синтаксис образования конструкций языка):

- машинные языки (computer language) - языки программирования, воспринимаемые аппаратной частью компьютера (машинные коды);
- машинно-ориентированные языки (computer-oriented language) - языки программирования, которые отражают структуру конкретного типа компьютера (ассемблеры);
- алгоритмические языки (algorithmic language) - языки программирования, не зависящие от архитектуры компьютера (Паскаль, Си, Фортран, Бейсик и др.);
- процедурно-ориентированные языки (procedure-oriented language) - языки программирования, где имеется возможность написания программы как совокупности процедур (подпрограмм);
- проблемно-ориентированные языки (universal programming language) - языки программирования, предназначенные для решения задач определенного класса (Лисп, Пролог, Симула и др.);
- интегрированные системы программирования.

Другой классификацией языков программирования является их деление на языки, ориентированные на реализацию основ структурного программирования, и объектно-ориентированные языки, поддерживающие понятие объектов и их свойств и методов обработки.

Программа, подготовленная на языке программирования, проходит этап трансляции, когда происходит преобразование исходного кода программы (source code) в объектный код (object code), который далее пригоден к обработке редактором связей. Редактор связей специальная программа, обеспечивающая построение загрузочного модуля (load module), пригодного к выполнению.

Трансляция может выполняться с использованием средств компиляторов (compiler) или интерпретаторов (interpreter). Компиляторы транслируют всю программу, но без ее выполнения. Интерпретаторы, в отличие от компиляторов, выполняют пооператорную обработку и выполнение программы.

Существуют специальные программы, предназначенные для трассировки и анализа выполнения программ, так называемые отладчики (debugger). Лучшие отладчики позволяют осуществить трассировку (отслеживание выполнения программы в пооператорном варианте), идентификацию места и вида ошибок в программе, наблюдение за изменением значений переменных, выражений и т.п. Для отладки и тестирования правильности работы программ создается база данных контрольного примера.

Более мощным средством разработки программ являются *системы программирования*.

Системы программирования (programming system) включают:

- компилятор;
- интегрированную среду разработчика программ;
- отладчик;
- средства оптимизации кода программ;
- набор библиотек (возможно с исходными текстами программ);
- редактор связей;
- сервисные средства (утилиты) для работы с библиотеками текстовыми и двоичными файлами;
- справочные системы;
- документатор исходного кода программы;
- систему поддержки и управления проектом программного комплекса.

Средства поддержки проектов - новый класс средств разработки программного обеспечения, предназначенный для:

- отслеживания изменений, выполненных разработчиками программ;
- поддержки версий программы с автоматической разностной изменений;
- получения статистики о ходе работ проекта.

Инструментальная среда пользователя представлена специальными средствами, встроенными в пакеты прикладных программ, такими, как:

- библиотека функций, процедур, объектов и методов обработки;
- макрокоманды;
- клавишные макросы; языковые макросы;
- программные модули-вставки; конструкторы экранных форм и отчетов;
- генераторы приложений; языки запросов высокого уровня;
- языки манипулирования данными; конструкторы меню и многое другое.

Средства отладки и тестирования программ предназначены для подготовки разработанной программы к промышленной эксплуатации.

Интегрированные среды разработки программ

Дальнейшим развитием локальных средств разработки программ, являются интегрированные программные среды разработчиков.

Основное назначение инструментария данного вида - повышение производительности труда программистов, автоматизация создания кодов программ, обеспечивающих интерфейс пользователя графического типа, разработка приложений для архитектуры клиент-сервер, запросов и отчетов.

РЕЦЕНЗИЯ

на рабочую программу учебной дисциплины
МДК.02.01 Технология разработки программного обеспечения
для специальности 09.02.07 Информационные системы и программирование

Рабочая программа учебной дисциплины МДК.02.01 «Технология разработки программного обеспечения» 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 экспертизы можно сделать заключение, что рабочая программа учебной дисциплины МДК.02.01 «Технология разработки программного обеспечения» по специальности 09.02.07 «Информационные системы и программирование» соответствует требованиям стандарта, профессиональным требованиям, а также современным требованиям рынка труда.

Технический директор
ООО «ТехноСтарт»

« » 20 г.



И.Г. Колодезный

РЕЦЕНЗИЯ

на рабочую программу учебной дисциплины
МДК.02.01 Технология разработки программного обеспечения
для специальности 09.02.07 Информационные системы и программирование

Рабочая программа учебной дисциплины МДК.02.01 «Технология разработки программного обеспечения» 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 экспертизы можно сделать заключение, что рабочая программа учебной дисциплины МДК.02.01 «Технология разработки программного обеспечения» 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