

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ
БУРЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ ДОРЖИ БАНАЗАРОВА
ИНСТИТУТ МАТЕМАТИКИ, ФИЗИКИ И КОМПЬЮТЕРНЫХ НАУК
КАФЕДРА ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ И ИНФОРМАТИКИ

Утверждена на заседании
Ученого совета ИМФКН
«__»_____ 202__ г.
Протокол № __

Рабочая программа дисциплины
Современные технологии разработки программного обеспечения

Направление подготовки / специальность
09.04.02 Информационные системы и технологии
Профиль
Проектирование, разработка и эксплуатация информационных систем

Квалификация (степень) выпускника
Магистр

Форма обучения
Очная

Улан-Удэ
2025

Пояснительная записка

Цели освоения дисциплины

Целью изучения курса "Современные технологии разработки программного обеспечения" является ознакомление магистрантов с современным состоянием и проблемами технологии разработки программного обеспечения; изучение новых подходов к построению программных комплексов, новых парадигм построения программного обеспечения

Место дисциплины в структуре образовательной программы

Данная дисциплина входит в раздел Б1.В для направления подготовки 09.04.02 Информационные системы и технологии. В дисциплине используются знания по программированию, архитектуре вычислительных систем.

Планируемые результаты обучения по дисциплине и индикаторы достижения компетенций.

УК-2. Способен управлять проектом на всех этапах его жизненного цикла

УК-2.2 разрабатывает концепцию проекта в рамках обозначенной проблемы: формулирует цель, задачи, обосновывает актуальность, значимость, ожидаемые результаты и возможные сферы их применения

УК-2.3 разрабатывает план реализации проекта с учетом возможных рисков реализации и возможностей их устранения, планирует необходимые ресурсы

ПК-3. Способен планировать коммуникации в проектах любого уровня сложности в области ИТ

ПК-3.1. Подбирает оптимальные инструменты и методы коммуникаций

ПК-3.3 Владеет современными инструментами и методами коммуникаций

В результате освоения дисциплины студент должен:

Знать:

жизненный цикл программ, технологии разработки программных комплексов, CASE-средства;

методы и алгоритмы объектно-ориентированного программирования;

средства информационных технологий, методы разработки программно-аппаратных комплексов объектов профессиональной деятельности;

специальные программные средства для отладки программного кода;

Уметь:

проводить оценку качества программных продуктов, технологии разработки программных комплексов;

использовать типовые программные продукты, ориентированные на решение научных, проектных и технологических задач

Владеть:

навыками объектно-ориентированного анализа и проектирования;

навыками проектирования программного обеспечения в современных средах разработки.

Планируемые результаты освоения образовательной программы:

Объем дисциплины в зачетных единицах с указанием количества часов, выделенных на контактную работу обучающихся с преподавателем и на самостоятельную работу обучающихся

Общая трудоемкость дисциплины составляет 6 зачетные единицы, 216 часа.

№	Название разделов дисциплины	Лекция	Самостоятельная работа	Лабораторная работа
Семестр 1		14	80	14

№	Название разделов дисциплины	Лекция	Самостоятельная работа	Лабораторная работа
1	Задача проектирования программных систем	2	20	2
2	Технологические средства разработки программного обеспечения	2	20	2
3	Процесс проектирования программного обеспечения	4	20	4
4	Методология объектно-ориентированного программирования	6	20	6
Семестр 2		0	96	12
1	Проектирование интерфейса		24	2
2	Методы отладки и тестирования программного обеспечения		20	4
3	Организация процесса проектирования программного обеспечения		26	4
4	Документирование и оценка качества программных продуктов		26	2

Тематическое планирование курса

Темы

Задача проектирования программных систем

Семестр 1

Задача проектирования программных систем

Лекция. 2 ч. Ведение в технологию программирования, программную инженерию. Жизненный цикл программного продукта. Процессы ЖЦ. Модели ЖЦ.

Самостоятельная работа. 20 ч. История технологии программирования. Основные этапы решения задач на ЭВМ; критерии качества программы; интерфейс пользователя: консольные и диалоговые программы, дружелюбность интерфейса пользователя; жизненный цикл программы, водопадная модель.

Лабораторная работа. 2(0) ч. Основные этапы решения задач на ЭВМ. Проектирование программных систем

Технологические средства разработки программного обеспечения

Семестр 1

Технологические средства разработки программного обеспечения

Лекция. 2(0) ч. Инструментальная среда разработки, средства поддержки проекта, отладчики.

Лабораторная работа. 2(0) ч. Знакомство с инструментальной средой разработки ПО

Самостоятельная работа. 20 ч. Машины Тьюринга. Модульность в программировании. Основные парадигмы программирования.

Процесс проектирования программного обеспечения

Семестр 1

Процесс проектирования программного обеспечения

Лекция. 4(0) ч. Использование декомпозиции и абстракции при проектировании ПО; специфики процедур и данных; декомпозиция системы. Методы проектирования структуры ПО; методы защиты программ и данных.

Лабораторная работа. 4(0) ч. Разработка и тестирование алгоритма

Самостоятельная работа. 20(0) ч. Процедурное программирование. Стандарты языка. Функции. Встраиваемые функции.

Методология объектно-ориентированного программирования

Семестр 1

Методология объектно-ориентированного программирования

Лекция. 6 ч. Понятия объекта, класса объектов. Основные понятия объектно-ориентированного программирования: инкапсуляция, наследование и полиморфизм. Классы. Функции-члены и данные-члены. Интерфейсы и реализация. Конструкторы и инициализация. Деструкторы и очистка. Наследование классов и производные классы. Наследование. Множественное наследование. Виртуальные базовые классы. Иерархия классов. Виртуальные функции. Полиморфизм. Абстрактные классы и чистые виртуальные функции. Предопределенные объекты и потоки.

Лабораторная работа. 6(0) ч. Конструирование класса Использование наследования для создания нового класса

Самостоятельная работа. 20 ч. Класс и объекты основы. Инкапсуляция. Класс как пользовательский курс. Связь класса со структурой. Конструктор и деструктор.

Проектирование интерфейса

Семестр 2

Проектирование интерфейса

Самостоятельная работа. 4(0) ч. Проектирование интерфейса с пользователем. Структуры диалога; поддержка пользователя; многооконные интерфейсы; примеры реализации интерфейсов с пользователем с использованием графических пакетов.

Лабораторная работа. 2(0) ч. Библиотеки C и C++ для разработки интерфейса пользователя, Библиотека Win32 API, приложение с интерфейсом одного (SDI) и многих (MDI) документов, обработка сообщений, карта сообщений, программирование основных элементов управления; библиотека Qt, иерархия классов, структура приложения, элементы управления, обработка сообщений, слоты, сигналы. Создание диалогового приложения на основе. Создание программ на Си++ с графическим интерфейсом пользователя

Самостоятельная работа. 20(0) ч. Венгерская нотация, типы данных, дескрипторы, сообщения и обработка сообщений, создание окон, основные функции, оконная процедура, контексты устройств, ресурсы приложения: диалоговая панель, меню, акселератор; библиотека MFC, иерархия классов, структура приложения, диалоговое приложение.

Методы отладки и тестирования программного обеспечения

Семестр 2

Методы отладки и тестирования программного обеспечения

Лабораторная работа. 2(0) ч. Категории программных ошибок. Типы тестов. Тестирование на этапе планирования. Тестирование на этапе проектирования. Тестирование "стеклянного ящика" на стадии кодирования. Регрессионное тестирование. Тестирование "черного ящика". Разработка тестов.

Лабораторная работа. 2(0) ч. Автономная отладка и тестирование программного модуля. Комплексная отладка и тестирование программного средства.

Самостоятельная работа. 20(0) ч. Структура файлов проекта и их взаимодействие. Заголовочный файл. Файл исходного кода. Разделение функции на файлы

Организация процесса проектирования программного обеспечения

Семестр 2

Организация процесса проектирования программного обеспечения

Лабораторная работа. 4(0) ч. Проектирование программного обеспечения.

Функциональность и надежность как обязательные критерии качества программного средства. Обеспечение завершенности программного средства. Защитное программирование и обеспечение устойчивости программного модуля. Виды защиты и обеспечение защищенности программного средства.

Самостоятельная работа. 6(0) ч. Работа с требованиями к ПО. Функциональные и пользовательские требования. Проект. Управление проектом и ресурсами проекта.

Самостоятельная работа. 20(0) ч. Программирование ПО. Трудность оценки программы. Проектирование и конструирование ПП. Уровни проектирования. Декомпозиция задачи на подсистемы.

Документирование и оценка качества программных продуктов

Семестр 2

Документирование и оценка качества программных продуктов

Лабораторная работа. 2(0) ч. Документация, создаваемая в процессе разработки программных средств. ЕСПД. Пользовательская документация программных средств.

Документация по сопровождению программных средств. Стандарт ISO 9126. Модель качества. Характеристики и субхарактеристики качества программного средства.

Самостоятельная работа. 6(0) ч. Метрики качества программного средства. Оценивание характеристик качества программных средств. Схемы алгоритмов, программ, данных и систем; состав эксплуатационных документов: руководство пользователя, программиста, системного программиста.

Самостоятельная работа. 20(0) ч. Стандарты ГОСТ и ISO, стандарты на программную документацию, система ГОСТов ЕСПД: ГОСТ 19.001-77 - 19.64-78; техническое задание, основные разделы; виды программ и программных документов; спецификация программного продукта; пояснительная записка; ГОСТ 19.701-90

БРС

Семестр	Контрольные точки	Баллы
1	Текущий контроль в разделе «Задача проектирования программных систем»	
	Защита проекта	60
1	Зачет	
	Тест	25
	Устный ответ на вопрос	15
Итого за семестр 1:		100
2	Текущий контроль в разделе «Методы отладки и тестирования программноого обеспечения»	
	Защита проекта	60

Семестр	Контрольные точки	Баллы
2	Зачет	
	Тест	25
	Устный ответ на вопрос	15
Итого за семестр 2:		100

Учебно-методическое и информационное обеспечение учебного процесса

Образовательные технологии (в том числе на занятиях, проводимых в интерактивных формах).

Теоретическая часть курса, общие вопросы технологий разработки программного обеспечения излагаются в лекционном курсе. Отдельные вопросы могут выноситься на самостоятельное изучение. Для приобретения навыков работы на ПК предназначены лабораторные занятия. При изучении дисциплины используются интерактивные формы занятий (лекция-дискуссия, защита рефератов) в объеме 10 часов.

- [Методич рек CPC.doc](#)

Режим доступа: https://my.bsu.ru/content/file/2/21/213/69325_metodich_rek_srs.doc

Учебно-методические материалы, в том числе методические указания для обучающихся по освоению дисциплины

Теоретическая часть курса, общие вопросы методики и технологий применения компьютерных средств излагаются преподавателем в лекционном курсе. Отдельные вопросы могут выноситься на самостоятельное изучение. Студент должен иметь в виду, что на лекциях преподаватель определяет такие вопросы и рекомендует необходимую для их изучения литературу, источники и др. ресурсы. Для успешного освоения курса необходимо внимательно фиксировать основные положения лекции, своевременно их усваивать, при необходимости самостоятельно прорабатывать, используя основную и дополнительную литературу.

Для приобретения навыков общения с ПК в процессе освоения инструментальных систем и отладки программ предназначены лабораторные занятия. Лабораторные занятия проводятся в специальных классах, оборудованных средствами вычислительной техники. На первом лабораторном занятии студенты получают инструктаж по технике безопасности при работе в классе и знакомятся с особенностями работы на конкретной вычислительной машине. Последующие лабораторные работы заключаются в освоении инструментальных систем и отладке программ решения типовых задач. Индивидуальные задания и методические указания к выполнению каждой последующей лабораторной работы студент получает, как правило, на предыдущем занятии. Подготовка к выполнению лабораторных работ осуществляется в часы самостоятельной работы. Студенты, не подготовившиеся к занятиям, к работе на компьютере не допускаются. По каждой выполненной лабораторной работе студент оформляет отчет по установленной форме.

Самостоятельные занятия под контролем преподавателя предназначены для самостоятельного изучения студентами тех разделов курса, по которым не предусмотрено чтение лекций, либо проводятся лекции обзорного характера. По усмотрению

преподавателя в часы индивидуальных занятий студентам может поручаться выполнение других заданий. Занятия проводятся с академической группой или с половиной группы в часы, установленные расписанием занятий. На занятиях студент должен иметь конспект лекций, учебную и справочную литературу, отдельную тетрадь для записей. Весь теоретический материал, изученный в процессе индивидуальных занятий, должен быть законспектирован.

Оценочные средства

По данной дисциплине разработаны оценочные средства, критерии их оценивания, а также методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций.

- [Методич рек СРС.doc](#)
- [ФОС. Современные технологии разработки программного обеспечения. Маги.docx](#)

Список литературы

Перечень основной и дополнительной литературы, необходимой для освоения дисциплины.

Основная

1. [Эффективное тестирование программного обеспечения](#)/Аниче М.. —Москва: ДМК Пресс, 2023. —370 с.
Режим доступа: <https://e.lanbook.com/book/314930>
2. [Основы тестирования и верификации программного обеспечения](#): учебное пособие для вузов/Старолетов С. М.. —Санкт-Петербург: Лань, 2023. —344 с.
Режим доступа: <https://e.lanbook.com/book/319445>
3. [Надежность и безопасность программного обеспечения](#): Учебное пособие для вузов/Казарин О. В., Шубинский И. Б.. —Москва: Юрайт, 2022. —342 с.
Режим доступа: <https://urait.ru/bcode/493262>

Дополнительная

1. [Языки программирования. Концепции и принципы](#)/Кауфман В.Ш.. —Москва: ДМК Пресс, 2010
Режим доступа: http://e.lanbook.com/books/element.php?pl1_cid=25&pl1_id=1270
2. [Стандартизация и разработка программных систем](#): учеб. пособие/В. Н. Гусятников, А. И. Безруков. —Москва: Финансы и статистика, 2010. —288 с.
Режим доступа: http://e.lanbook.com/books/element.php?pl1_cid=25&pl1_id=5321

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

<http://www.intuit.ru/departament/informatics/intinfo/>- Введение в информатику. Курс Интернет-университета информационных технологий

<http://www.junior.ru/wwwexam/> - Информатика и информационные технологии - web-конспект и тесты

<http://www.intuit.ru/departament/se/pinform/>- Практическая информатика. Курс Интернет-университета информационных технологий

<http://www.ict.edu.ru> - Информационно-коммуникационные технологии в образовании: федеральный образовательный портал

<http://www.computer-museum.ru> - Виртуальный компьютерный музей
<http://www.problems.ru/inf> - Задачи по информатике
<http://iit.metodist.ru> Информатика - и информационные технологии: сайт лаборатории информатики МИОО
<http://www.intuit.ru> - Интернет-университет информационных технологий (ИНТУИТ.ру)
<http://www.edu-it.ru> - ИТ-образование в России: сайт открытого е-консорциума
<http://edu.of.ru> - Конструктор образовательных сайтов (проект Российского общеобразовательного портала)
<http://labinfo.ioso.ru> - Лаборатория обучения информатике Института содержания и методов обучения РАО
<http://test.specialist.ru> - Онлайн-тестирование и сертификация по информационным технологиям
<http://www.firststeps.ru> - Первые шаги: уроки программирования
<http://algotlist.manual.ru> - Проект AlgoList: алгоритмы и, методы
<http://alglib.sources.ru> - Проект Alglib.ru: библиотека алгоритмов

Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения и информационных справочных систем (при необходимости)

Microsoft Visual Studio 2010, 2012, 2013, 2016 Ultimate
Office Standard 2007

Windows 7 Корпоративная

Личный кабинет преподавателя или студента БГУ <http://my.bsu.ru>

Федеральное интернет-тестирование: проекты «Интернет-тренажеры в сфере профессионального образования» и «Федеральный интернет-экзамен в сфере профессионального образования»

Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине

Аудитория 0417

Корпус:главный

Назначение аудитории:учебная аудитория для проведения занятий лекционного типа, занятий семинарского типа, курсового проектирования (выполнения курсовых работ), групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации

Число посадочных мест:19

Площадь (кв. м):55.4

Оборудование:персональный компьютер - 12 шт.

Аудитория 0419

Корпус:главный

Назначение аудитории:учебная аудитория для проведения занятий лекционного типа, занятий семинарского типа, курсового проектирования (выполнения курсовых работ), групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации

Число посадочных мест:30

Площадь (кв. м):55.2

Оборудование:

персональный компьютер - 13 шт.

интерактивная система - 1 шт.

принтер - 1 шт.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
ФГБОУ ВО «БУРЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ ДОРЖИ БАНЗАРОВА»

Фонд оценочных средств по дисциплине
Современные технологии разработки программного обеспечения

Направление подготовки/ специальность
09.04.02 Информационные системы и технологии

Профиль подготовки /специализация
Проектирование, разработка и эксплуатация информационных систем

Квалификация (степень) выпускника

магистр

Форма обучения

очная

Паспорт фонда оценочных средств

УК-2.2. Разрабатывает концепцию проекта в рамках обозначенной проблемы: формулирует цель, задачи, обосновывает актуальность, значимость, ожидаемые результаты и возможные сферы их применения

УК-2.3. Разрабатывает план реализации проекта с учетом возможных рисков реализации и возможностей их устранения, планирует необходимые ресурсы

ПК-3.1. Создает формальные методики оценки пользовательского интерфейса

ПК-3.3. Участвует в разработке рекомендаций по оптимизации интерфейсных решений программных продуктов и аппаратных средств

№	Контролируемые разделы, темы, модули	Формируемые компетенции	Этапы формирования	Оценочные средства	
				Вид	Количество
1.	Технологические средства разработки программного обеспечения	УК-2.2 УК-2.3 ПК-3.1	1 семестр	Защита проекта	1
2.	Методология объектно-ориентированного программирования	УК-2.2 УК-2.3 ПК-3.3	2 семестр	Защита проекта	1
Всего:				1	2

Описание показателей и критериев оценивания уровня приобретенных компетенций на различных этапах их формирования

Результаты обучения	Показатели оценивания	Критерии сформированности	Шкала оценивания
Знает: базовые понятия о жизненном цикле программ, технологии разработки программных комплексов, CASE-средства; методы и алгоритмы объектно-ориентированного программирования; средства информационных технологий, методы разработки программно-аппаратных комплексов объектов профессиональной деятельности;	Пороговый уровень (как обязательный для всех студентов)	Знает: базовые понятия о жизненном цикле программ, технологии разработки программных комплексов, CASE-средства; Умеет проводить оценку качества программных продуктов; Владеет: навыками объектно-ориентированного анализа	60-69 баллов
	Базовый уровень	<i>помимо удовлетворения критериям порогового уровня:</i> Знает: методы и алгоритмы объектно-ориентированного программирования; Умеет: применять технологии разработки программных комплексов; Владеет: навыками объектно-ориентированного анализа и проектирования	70-84 баллов
	Высокий уровень	<i>помимо удовлетворения критериям порогового и базового уровня:</i> Знает: методы разработки программно-	85-100 баллов

специальные программные средства для отладки программного кода; Умеет: проводить оценку качества программных продуктов, применять технологии разработки программных комплексов; использовать типовые программные продукты, ориентированные на решение научных, проектных и технологических задач Владеет: навыками объектно-ориентированного анализа и проектирования; навыками проектирования программного обеспечения в современных средах разработки.		аппаратных комплексов объектов профессиональной деятельности; Умеет: использовать типовые программные продукты, ориентированные на решение научных, проектных и технологических задач; Владеет: навыками проектирования программного обеспечения в современных средах разработки	
--	--	--	--

Балльно-рейтинговая система

по дисциплине «Современные технологии разработки программного обеспечения»

Общая максимальная сумма баллов, которую студент может набрать по дисциплине в течение семестра – 100 баллов: 60 баллов текущий контроль и рубежный контроль + 40 баллов зачет/экзамен (итоговый контроль);

– общая максимальная сумма баллов, которую студент может набрать в течение семестра за выполнение всех видов работ во время аудиторных и внеаудиторных занятий, активность и посещаемость, должна быть равна 60 баллам;

– минимальная сумма баллов, при которой студент допускается к зачету/экзамену (итоговому контролю), равна 36 баллам (60% от 60 баллов);

– минимальная сумма баллов, при которой студент получает положительную итоговую оценку по дисциплине равна 60 баллам (60% от 100 баллов).

Связь между четырехбалльной и столбальной системами оценки качества обучения студентов

Оценка	Буквенный эквивалент оценки	Рейтинговые баллы
Отлично	A+	95-100
	A	90-94

	A-	85-89
Хорошо	B+	80-84
	B	75-79
	B-	70-74
Удовлетворительно	C+	67-69
	C	64-66
	C-	60-63
Неудовлетворительно	D	40-59
–	F	<40
Зачтено	S	60-100
Не зачтено	U	<60

Оценочные средства и критерии их оценки

Примерные вопросы к зачету(1 семестр):

1. Стратегии разработки ПО и модели жизненного цикла. Каскадная модель, спиральная модель и инкрементная стратегии разработки ПО. Перечень, последовательность и содержимое этапов разработки по методологиям. Привести примеры итераций, задач, макетов и моделей.
2. Стратегии разработки ПО и модели жизненного цикла. Макетирование и модель быстрой разработки RAD. Перечень, последовательность и содержимое этапов разработки по методологиям. Примеры итераций, задач, макетов и моделей.
3. Тяжеловесные и облегченные процессы разработки ПО. Экстремальное программирование (XP-процесс). Перечень, последовательность и содержимое этапов разработки по методологиям. Примеры итераций, задач, макетов и моделей.
4. Стратегии разработки ПО и модели жизненного цикла. Agile (Scrum) процесс и компонентно-ориентированная модель. Перечень, последовательность и содержимое этапов разработки по методологиям. Примеры итераций, задач, макетов и моделей.
5. Руководство программным проектом. Планирование проекта. Структурная декомпозиция работ. Ресурсы и риски. Распределение задач и ресурсов. Вехи. Привести определения и примеры.
6. Руководство программным проектом. Планирование проекта. Анализ и управление рисками. Типы рисков. Календари ресурсов. Привести определения и примеры.
7. Оценка бюджета проекта: экспертная, по аналогии, алгоритмическая, на основе графика выполнения. Точность оценки. Оценка затрат, длительности и стоимости разработки. Метрический базис. Привести определения и примеры.
8. Оценка затрат, длительности и стоимости разработки. Оценка размера проекта на основе LOC, FP или OP. Привести определения и примеры.
9. Оценка затрат, длительности и стоимости разработки. Размерно-ориентированные и функционально-ориентированные метрики, их соотношение. Привести определения и примеры.
10. Вычисление затрат, длительности и стоимости разработки на основе моделей СОСОМО. Модели композиции приложения, раннего этапа проектирования и этапа пост-архитектуры. Привести определения и примеры.
11. Оценка затрат, длительности и стоимости разработки. Анализ чувствительности программного проекта с применением модели СОСОМО. Цель и методы выполнения. Привести определения и примеры.

Критерии оценки теоретической части:

14-15 баллов выставляется студенту, если он

- Четко знает стратегии разработки программных средств и систем, умеет создавать модели жизненного цикла, овладел умением строить диаграммы моделирования в языке UML.
- Дает четкий и правильный ответ, выявляющий понимание учебного материала и характеризующий прочные знания, излагает материал в логической последовательности с использованием принятой терминологии;

- Ошибок не делает, но допускает оговорки по невнимательности при работе с программными продуктами, которые легко исправляет по требованию преподавателя;
 - Ответ логичен, последователен, технически грамотен.
- 11-13 баллов выставляется студенту, если он*
- Овладел программным материалом, ориентируется в базовых концепциях разработки программных средств и систем, умеет создавать модели жизненного цикла с небольшим затруднением, но диаграммы моделирования в языке UML;
 - Дает правильный ответ в определенной логической последовательности;
- 9-10 баллов выставляется студенту, если он*
- Основной программный материал знает нетвердо, но большинство изученных понятий и обозначений усвоил;
 - Ответ дает неполный, построенный несвязно, но выявивший общее понимание вопросов;
- 0-8 баллов выставляется студенту, если он*
- Обнаруживает незнание или непонимание большей или наиболее важной части учебного материала;
 - Ответы строит несвязно, допускает существенные ошибки, которые не может исправить даже с помощью преподавателя.

Примерные вопросы к зачету(2 семестр):

1. Проектирование программных систем. Декомпозиция подсистем на модули.
2. Модуль и модульность. Виды связности модуля. Виды сцепление модулей. Привести определения и примеры.
3. Оценка качества программных систем. Функциональные метрики программных систем. Сложность модуля. Сложность программной системы.
4. Проектирование программных систем. Модели архитектуры: хранилище данных, клиент-сервер, трехуровневая модель, многоуровневая модель, каналы и фильтры, компонентная и каркасы. Привести определения и примеры.
5. Проектирование программных систем. Структурирование системы.
6. Определение моделей архитектуры, управления и интерфейсов взаимодействия. Привести определения и примеры.
7. Структурные шаблоны. MVC: подсистемы, их назначение и схема взаимодействия. Требования и особенности. Реализация обратного вызова. Привести примеры иллюстрирующих диаграмм классов и взаимодействия.
8. Структурные шаблоны. PCMEF: подсистемы, их назначение и схема взаимодействия. Требования и особенности. Делегирование. Привести примеры иллюстрирующих диаграмм классов и взаимодействия.
9. Структурные шаблоны. PCMEF+: подсистемы, их назначение и схема взаимодействия. Требования и особенности. Привести примеры иллюстрирующих диаграмм классов и взаимодействия.
10. Паттерны проектирования. Назначение и принцип описания. Типы и виды паттернов (бизнес-логика, БД, Gof). Параметры паттернов. Привести определения и примеры.
11. Паттерны для реализации бизнес-логики. Сценарий транзакций и модель предметной области. Назначение, особенности реализации и область применения. Примеры диаграмм классов и взаимодействия для указанных паттернов.

Критерии оценки теоретической части:

14-15 баллов выставляется студенту, если он

- Четко знает виды связности модуля, умеет применять паттерны проектирования, овладел умением проводить декомпозицию подсистем на модули.
- Дает четкий и правильный ответ, выявляющий понимание учебного материала и характеризующий прочные знания, излагает материал в логической последовательности с использованием принятой терминологии;
- Ошибок не делает, но допускает оговорки по невнимательности при работе с программными продуктами, которые легко исправляет по требованию преподавателя;
- Ответ логичен, последователен, технически грамотен.

11-13 баллов выставляется студенту, если он

- Овладел программным материалом, ориентируется в паттернах проектирования, умеет проводить декомпозицию подсистем на модули с небольшим затруднением, но знает виды связности модуля;

- Дает правильный ответ в определенной логической последовательности;

9-10 баллов выставляется студенту, если он

- Основной программный материал знает нетвердо, но большинство изученных понятий и обозначений усвоил;

- Ответ дает неполный, построенный несвязно, но выявивший общее понимание вопросов;

0-8 баллов выставляется студенту, если он

- Обнаруживает незнание или непонимание большей или наиболее важной части учебного материала;

- Ответы строит несвязно, допускает существенные ошибки, которые не может исправить даже с помощью преподавателя.

Примерные тестовые задания к зачету(1 семестр):

Тест №1

Вопрос 1

Чёткая последовательность действий, выполнение которой даёт какой-то заранее известный результат, называется ...

Варианты ответов

- Алгоритм
- Инструкция
- Программа
- Рецепт

Вопрос 2

Какое из перечисленных свойств НЕ является свойством алгоритма?

Варианты ответов

- Дискретность
- Определенность
- Массовость
- Вариативность

Вопрос 3

С помощью языка программирования создается текст, описывающий ранее составленный алгоритм. Чтобы получить работающую программу, надо этот текст перевести в последовательность команд процессора, что выполняется при помощи специальных программ, которые называются ...

Варианты ответов

- компиляторами
- трансляторами
- интерпретаторами

Вопрос 4

... читает всю программу целиком, делает ее перевод и создает законченный вариант программы на машинном языке, который затем и выполняется. Результат работы ... — бинарный исполняемый файл.

Варианты ответов

- Компилятор
- Интерпретатор
- Конвертер

Вопрос 5

Первый этап в жизненном цикле программы:

Варианты ответов

- Тестирование
- Формулирование требований
- Анализ требований

- Проектирование

Вопрос 6

Какой из перечисленных этапов жизненного цикла программы является НЕОБЯЗАТЕЛЬНЫМ?

Варианты ответов

- Разработка
- Тестирование
- Оптимизация
- Анализ требований

Вопрос 7

В стадии разработки программы НЕ входит:

Варианты ответов

- Автоматизация программирования
- Постановка задачи
- Эскизный проект
- Тестирование

Вопрос 8

Самый важный критерий качества программы:

Варианты ответов

- Надежность
- Эффективность
- Быстродействие
- Работоспособность

Вопрос 9

Какой из перечисленных критериев качества программы наиболее важный?

Варианты ответов

- Надежность
- Удобный интерфейс
- Быстродействие
- Эстетическая привлекательность

Вопрос 10

Какой способ оценки надежности программы единственный верный?

Варианты ответов

- Тестирование
- Сравнение с аналогами
- Трассировка
- Оптимизация

Вопрос 11

... программных систем - это период от момента возникновения идеи создания программного обеспечения до вывода программной системы из эксплуатации.

Вопрос 12

... в программировании - это процесс оценки работы программного обеспечения с целью обнаружения ошибок или несоответствия требованиям.

Вопрос 13

Как называется окно, в котором можно набирать команды и получать текстовые ответы от программы или системы?

Варианты ответов

- Консоль
- Окно Windows
- Диалоговое окно
- Блокнот

Вопрос 14

Какую команду необходимо ввести в консоль, что получить на выход сообщение "Hello world"?

Варианты ответов

- Console.WriteLine("Hello world");
- ping Hello world
- Console.WriteLine("Hello world")
- WriteLine("Hello world");

Вопрос 15

В рамках какой процедуры отвечают на следующие вопросы:

1. Отвечает ли система общим и бизнес-целям организации-заказчика и организации-разработчика?
2. Можно ли реализовать систему, используя существующие на данный момент технологии и не выходя за пределы заданной стоимости?
3. Можно ли объединить систему с другими системами, которые уже эксплуатируются?

Варианты ответов

- Анализ осуществимости
- Тестирование ПО
- Написание плана разработки программного продукта

Вопрос 16

Какой из приведенных пунктов относится к этапу программирования?

Варианты ответов

- Написание кода программы
- Разработка интерфейса
- Создание дизайн-проекта
- Проектирование

Вопрос 17

На языке программирования составляется:

Варианты ответов

- Исходный код
- Исполняемый код
- Объектный код
- Алгоритм

Вопрос 18

Правила, которым должна следовать программа это:

Варианты ответов

- Структура
- Алгоритм
- Спецификация
- Состав информации

Вопрос 19

Можно ли внутри цикла поместить еще один цикл?

Варианты ответов

- Нет
- Да

Вопрос 20

Что выполняется раньше?

Варианты ответов

- Выбор языка программирования
- Написание исходного кода
- Разработка алгоритма
- Компиляция

Критерии оценки:

- оценка «отлично» (23-25 баллов) выставляется студенту, если количество правильных ответов >90%;
- оценка «хорошо» (20-22 баллов): 76%-90%;
- оценка «удовлетворительно» (15-19 балла): -50% - 75%;
- оценка «неудовлетворительно» (0-14 баллов): <50%.

Примерные тестовые задания к зачету(2 семестр):

Тест №2

1 вопрос:

Укажите последовательность этапов ЖЦ ПС

Последовательность понятий:

- [3] Проектирование
- [2] Анализ требований и определение спецификаций
- [4] Реализация
- [5] Сопровождение
- [1] Постановка задачи

2 вопрос:

Первой моделью ЖЦ ПС является

Варианты ответа:

- ☒ каскадная
- ☐ спиральная
- ☐ с промежуточным контролем
- ☐ сопровождение

3 вопрос:

Какая модель ЖЦ ПО предъявляет высокие требования к наглядности описания разрабатываемого программного средства

Варианты ответа:

- ☐ RAD-технология
- ☐ COM-технология
- ☐ GPRS
- ☒ CASE-технология

4 вопрос:

Какая модель ЖЦ предполагает быстрое получение версий разрабатываемого программного обеспечения

Варианты ответа:

- ☐ CASE-технология
- ☒ RAD-технология
- ☐ COM-технология
- ☐ CMM

5 вопрос:

Автономно компилируемая программная единица

Ответ:

модуль

6 вопрос:

При каком методе программирования используются три вида вычислительного процесса: линейный, разветвленный, циклический

Варианты ответа:

- ☒ структурное
- ☐ объектно-ориентированное
- ☐ процедурное
- ☐ модульное

7 вопрос:

Формализованное текстовое описание алгоритма например, Выполнять <действие> До <условие>)

Ответ:

псевдокод

8 вопрос:

Программы и системы, ориентированные на решение конкретных пользовательских задач

Варианты ответа:

- ☐ системные
- ☒ прикладные
- ☐ оболочки
- ☐ общего назначения

9 вопрос:

Свойство программного продукта, обеспечивающее правильность работы при любых допустимых данных и защиту от неправильных данных

Варианты ответа:

- ☐ адаптируемость
- ☐ правильность
- ☒ универсальность
- ☐ точность результатов

10 вопрос:

Адресованный компьютеру набор инструкций, точно описывающий последовательность действий, которые необходимо выполнить

Ответ:

программа

11 вопрос:

Программное обеспечение, представляющее собой организованную совокупность подсистем (программ), позволяющих решать широкий класс задач из некоторой прикладной области

Варианты ответа:

- ☐ пакет программ
- ☐ программный комплекс
- ☒ программная система
- ☐ программа

12 вопрос:

Вид пользовательского интерфейса, реализующий работу принципом иерархической структуры

Варианты ответа:

- ☐ прямое манипулирование
- ☒ меню
- ☐ свободная навигация
- ☐ примитивный

13 вопрос:

Языки программирования, являющиеся частью профессиональных сред, характеризующиеся узкой направленностью для создания конкретных типов программного обеспечения

Варианты ответа:

- ☐ языки низкого уровня
- ☒ специализированные языки разработчика
- ☐ специализированные языки пользователя

14 вопрос:

Стандартный язык описания разработки программных продуктов с использованием объектного подхода

Варианты ответа:

- ☒ UML
- ☐ XML
- ☐ CASE-средства
- ☐ CMM

15 вопрос:

Порция информации, участвующая в диалоговом обмене

Ответ:

сообщение

16 вопрос:

Процесс выполнения программы, целью которого является выявление ошибок

Варианты ответа:

- ☐ отладка
- ☒ тестирование
- ☐ компиляция
- ☐ интерпретация

17 вопрос:

Процесс локализации и исправления ошибок, обнаруженных при тестировании

Ответ:

отладка

18 вопрос:

Система точно сформулированных правил, определяющая процесс преобразования исходных данных в результат за конечное число шагов

Варианты ответа:

- ☐ внешнее описание
- ☒ алгоритм
- ☐ информационный поток
- ☐ программный комплекс

19 вопрос:

Поддержка работоспособности программы (переход на новые версии, внесение изменений и т.д.)

Варианты ответа:

- ☐ отладка
- ☒ сопровождение
- ☐ компиляция
- ☐ программирование

20 вопрос:

Используется для обеспечения процесса разработки программ, включает специализированное программное обеспечение, поддерживающее все технологические этапы процесса проектирования, программирования, отладки и тестирования создаваемых программ

Варианты ответа:

- ☐ системное программное обеспечение
- ☒ инструментарий технологии программирования
- ☐ прикладное программное обеспечение

21 вопрос:

Транслирует программу без её выполнения

Варианты ответа:

- ☒ компилятор
- ☐ транслятор
- ☐ отладчик
- ☐ интерпретатор

22 вопрос:

В технологии программирования акцент делается на

Варианты ответа:

- ☒ изучение технологических процессов разработки программного средства и порядке их прохождения
- ☐ разработку детального внешнего описания разрабатываемого программного средства
- ☐ создание качественного программного средства
- ☐ изучение инструментальных средств разработки программных средств

23 вопрос:

К линейным структурам данных с последовательным распределением относятся

Варианты ответа:

- ☐ вектор
- ☐ массив
- ☐ запись
- ☐ двусвязный список

24 вопрос:

Единственно верным сцеплением программных модулей является

Варианты ответа:

- ☐ сцепление по содержимому
- ☐ сцепление по образцу
- ☒ сцепление по данным
- ☐ сцепление по внешним ссылкам

25 вопрос:

Характеристикой программного модуля, отражающей независимость от предыстории обращений к нему, является

Варианты ответа:

- ☒ рутинность
- ☐ прочность
- ☐ сцепление
- ☐ размер

26 вопрос:

Метод разработки программ при модульном программировании, когда реализация начинается с головного модуля, затем программируются модули уровнем ниже и т.д.

Варианты ответа:

- ☐ восходящая разработка
- ☒ нисходящая разработка
- ☐ архитектурный подход
- ☐ структурный подход

27 вопрос:

Словарь терминов внешнего описания предназначен для

Варианты ответа:

- ☐ формализованного описания алгоритма
- ☒ повышения степени понимания предметной области
- ☐ хранения основных показателей качества программного средства
- ☐ упрощения процесса тестирования программного средства

28 вопрос:

Какая модель отражает функциональную диаграмму

Варианты ответа:

- ☐ DFD
- ☐ ERD
- ☒ SADT
- ☐ STD

29 вопрос:

Для обозначения множества объектов, имеющих одинаковую структуру, поведение и отношения с другими объектами, используется

Варианты ответа:

- ☒ класс
- ☐ метод
- ☐ событие
- ☐ DFD-диаграмма

30 вопрос:

Какая модель ЖЦ соответствует CASE-технологии?

Варианты ответа:

- ☐ каскадная
- ☒ спиральная
- ☐ RAD
- ☐ модель с промежуточным контролем

31 вопрос:

При каком тестировании проверяется минимально возможный компонент (класс или функция)?

Варианты ответа:

- ☐ интеграционное
- ☐ альфа-тестирование
- ☒ модульное
- ☐ "черного ящика"

32 вопрос:

При каком тестировании проверяет, есть ли проблемы в интерфейсах или во взаимодействии между компонентами системы?

Варианты ответа:

- ☒ интеграционное
- ☐ системное
- ☐ модульное

☐ отладка

33 вопрос:

При каком тестировании разработчик теста имеет доступ к исходному коду и может вносить изменения в код?

Варианты ответа:

☐ модульное

☒ "белого ящика"

☐ "черного ящика"

☐ интеграционное

34 вопрос:

Какие ошибки "ловятся" на стадии компиляции?

Варианты ответа:

☐ многопользовательского доступа

☒ синтаксиса

☐ инструментария

☐ логики взаимосвязи модулей

35 вопрос:

Какой раздел не входит в руководство пользователя?

Варианты ответа:

☐ инструкция по работе

☐ общие сведения о программе

☐ описание установки

☒ методика испытаний

36 вопрос:

Какие языки позволяют осуществлять программирование машинными командами?

Варианты ответа:

☒ низкого уровня

☐ специализированные языки пользователя

☐ высокого уровня

☐ специализированные языки разработчика

37 вопрос:

Метод защиты информации шифрованием от несанкционированного прочтения

Варианты ответа:

☐ правовой

☐ лицензирование

☒ криптография

☐ обфускация

38 вопрос:

Как называются тесты, составляющиеся исходя из текста программы?

Варианты ответа:

☐ функциональные

☒ структурные

☐ эффективные

☐ специализированные

39 вопрос:

Что не является характеристикой программного модуля?

Варианты ответа:

☐ один вход и один выход

☐ функциональная завершенность

☐ логическая независимость

☒ высокая квалификация разработчика

40 вопрос:

Многопользовательскую архитектуру реализуют системы, построенные по принципу

Варианты ответа:

☐ пакет программ

☐ локальная сеть

☒ клиент-сервер

[] программная система

Критерии оценки:

- оценка «отлично» (23-25 баллов) выставляется студенту, если количество правильных ответов >90%;
- оценка «хорошо» (20-22 баллов): 76%-90%;
- оценка «удовлетворительно» (15-19 балла): -50% - 75%;
- оценка «неудовлетворительно» (0-14 баллов): <50%.

Примерные темы проектов(1 семестр)

1. Реализовать приложение командной строки, получающее от пользователя два натуральных числа и выводящее в стандартный поток вывода десятичные представления всех простых чисел, расположенных между ними, по одному на строку. Приложение должно иметь модульную архитектуру, в которой процедура получения списка простых чисел находится в отдельном модуле и не содержит взаимодействия с пользователем, так, чтобы его можно было использовать без модификации при реализации приложения с графическим интерфейсом.

2. Реализовать приложение с оконным графическим интерфейсом, позволяющее ввести значения двух натуральных чисел и выводящее список десятичных представлений всех простых чисел, расположенных между ними.

3. Создать для проекта графического приложения локальный репозиторий системы управления версиями Git. Настроить параметры user.name и user.email репозитория. Значение параметра user.name установить соответствующим фамилии студента. Настроить параметры фиксации таким образом, чтобы в репозиторий помещались только файлы проекта (исключив настройки среды, результаты сборки и другие вспомогательные файлы). Зафиксировать проект графического приложения в созданном репозитории. Реализовать в графическом приложении проверку того, что введенные пользователем данные являются десятичными представлениями чисел. При несоответствии должно выводиться диалоговое окно с сообщением, указывающим причину ошибки, и то, какой параметр введен неверно.

4. На основе модуля обработки данных создать библиотеку (пакет, сборку или другую аналогичную единицу используемого языка программирования). Модифицировать проекты приложений так, чтобы для обработки данных использовалась созданная библиотека. Средствами интегрированной среды разработки сделать проекты приложений зависимыми от проекта библиотеки так, чтобы при их сборке и запуске проверялась актуальность библиотеки и, при необходимости, выполнялось её обновление. Зафиксировать изменения проектов приложений в соответствующих репозиториях.

5. Настроить автоматическую стилизацию кода для проектов приложений и библиотеки. В проектах приложений параметры стилизации должны быть зафиксированы в соответствующих репозиториях системы управления версиями так, чтобы они применялись автоматически в каждой рабочей копии. Настроить обработчики репозитория проектов приложений таким образом, чтобы стилизация автоматически применялась перед каждой фиксацией ко всем файлам исходного кода.

Примерные темы проектов(2 семестр)

1. Разработка приложения для торгового предприятия
2. Разработка приложения для кредитного отдела банка
3. Разработка приложения для гостиницы
4. Разработка приложения для авторемонтной мастерской
5. Разработка приложения для автосалона
6. Разработка приложения для агентства недвижимости
7. Разработка приложения для склада
8. Разработка приложения для учета расходов семьи

9. Разработка приложения для организации делопроизводства
10. Разработка приложения для рекламного агентства
11. Разработка приложения для службы поддержки
12. Разработка приложения для кадровой службы организации
13. Разработка приложения для туристического агентства
14. Разработка приложения компьютерного магазина
15. Разработка приложения для страховой компании
16. Разработка приложения для кафедры вуза
17. Разработка приложения для мебельного магазина
18. Разработка приложения для книжного магазина
19. Разработка приложения для учета договоров в организации
20. Разработка приложения для строительной компании
21. Разработка приложения по взаимодействию с клиентами организации
22. Разработка приложения для взаимодействия с заказчиками в организации
23. Разработка приложения для салона красоты
24. Разработка приложения сервисного центра по ремонту компьютерной техники
25. Разработка приложения для транспортной компании
26. Разработка приложения для менеджера по продажам
27. Разработка приложения для учета компьютерной техники и программного обеспечения в организации
28. Разработка приложения для организации документооборота
29. Разработка приложения для управляющей компании ЖКХ
30. Разработка приложения для салона сотовой связи

Критерии оценки:

оценка «отлично» (56-60 баллов) выставляется студенту, если

- правильно обоснована стратегия разработки программного приложения, создана UML диаграмма, разработанное программное приложение функционирует без сбоев;
- продемонстрировано усвоение изученного учебного материала, сформированность и устойчивость используемых умений и навыков;

оценка «хорошо» (45-55 баллов) выставляется студенту, если

- правильно обоснована стратегия разработки программного приложения, создана UML диаграмма, разработанное программное приложение в некоторых случаях работает некорректно;
- продемонстрировано усвоение изученного учебного материала;

оценка «удовлетворительно» (31-44 баллов) выставляется студенту, если он

- UML диаграмма построена с недочетами, разработанное программное приложение в некоторых случаях работает некорректно;
- продемонстрировал недостаточную сформированность основных умений и навыков в применении правил на практике;

оценка «неудовлетворительно» (менее 30 баллов) выставляется студенту, если он

- разработанное программное приложение не отвечает поставленным требованиям либо не функционирует;
- обнаружено незнание или непонимание большей или наиболее важной части учебного материала;
- допущены грубые ошибки при использовании изученных правил и приемов.