

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ЛУГАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ ВЛАДИМИРА ДАЛЯ»

Факультет компьютерных систем и информационных технологий

Кафедра информатики и программной инженерии

УТВЕРЖДАЮ

Декан факультета Компьютерных
систем и информационных технологий

Кочевский А.А.

« 19 » 04 2023 г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
по учебной дисциплине

Современные парадигмы программирования

(наименование учебной дисциплины, практики)

09.03.03 Прикладная информатика

(код и наименование направления подготовки (специальности))

09.03. 03.01 Прикладная информатика в экономике

(наименование профиля подготовки (специальности, магистерской программы); при отсутствии ставится прочерк)

Разработчик (разработчики):

старший преподаватель _____ Сычева Л.Ф.

ФОС рассмотрен и одобрен на заседании кафедры информатики и
программной инженерии

от 18 апреля 2023 г., протокол № 17

Заведующий кафедрой _____

(подпись)

Кочевский А.А.

Луганск – 2023 г.

Паспорт
фонда оценочных средств по учебной дисциплине
«Современные парадигмы программирования»
Перечень компетенций (элементов компетенций), формируемых в
результате освоения учебной дисциплины (модуля) или практики

№ п/п	Код контролируемой компетенции	Формулировка контролируемой компетенции	Контролируемые темы учебной дисциплины, практики	Этапы формирования (семестр изучения)
1.	ПК-2	способность осуществлять проектирование программного обеспечения ИС и разрабатывать техническую документацию на его компоненты	Тема 1. Стили и языки программирования. Тема 2. Парадигмы программирования. Тема 3. Языки низкого уровня. Тема 4. Языки высокого уровня. Тема 5. Функциональное программирование. Тема 6. Логическое программирование. Тема 7. Объектно-ориентированное программирование. Тема 8. Параллельное программирование.	7

Показатели и критерии оценивания компетенций, описание шкал оценивания

№ п/п	Код контролируемой компетенции	Показатель оценивания (знания, умения, навыки)	Контролируемые темы учебной дисциплины	Наименование оценочного средства
1.	ПК-2	ПК-2.1. Способен использовать современные языки и системы программирования, технологии проектирования программного обеспечения ПК-2.2. Способен сформулировать требования к разрабатываемому программному	Тема 1. Стили и языки программирования. Тема 2. Парадигмы программирования. Тема 3. Языки низкого уровня. Тема 4. Языки высокого уровня. Тема 5. Функциональное программирование. Тема 6. Логическое программирование.	Вопросы для обсуждения (в виде докладов и сообщений), лабораторные работы, контрольные работы

		обеспечению, выполнить его реализацию и оформить техническую документацию на его компоненты ПК-2.3. Способен осуществлять проектирование программного обеспечения конкретной ИС и разработку технической документации на ее компоненты	Тема 7. Объектно-ориентированное программирование. Тема 8. Параллельное программирование.	
--	--	--	---	--

Фонды оценочных средств по дисциплине «Современные парадигмы программирования»

Вопросы для обсуждения (в виде докладов и сообщений):

1. Абдуктивное логическое программирование
2. Металогическое программирование
3. Логическое программирование в ограничениях
4. Параллельное логическое программирование (FGCS – японский проект 5-го поколения компьютерных систем)
5. Индуктивное логическое программирование
6. Линейное логическое программирование
7. Объектно-ориентированное логическое программирование;
8. Транзакционное логическое программирование.
9. Регистры абстрактной машины
10. Категории команд абстрактной машины
11. Реализационная прагмати
12. Парадигматическая специфика
13. Средства представления выражений с функциями;
14. Арифметические функции;
15. Неявные циклы форматного ввода-вывода векторов и массивов;
16. Вычисляемые передачи управления;
17. Сопрограммы – многовходовые подпрограммы.
18. Lisp поддерживает работу: с таблицей свойств атомов, фактически поддерживающей статическую память с побочным эффектом;
19. Императивно-процедурное программирование в форме Prog.
20. Глобальные определения атомов в терминах свойств атомов.

21. Псевдо-атомы для работы со скалярами разных типов и форматов.
22. Мультиоперации над числами и строками.
23. Псевдо-функции для взаимодействия с операционной системой.
24. Расширение системы программирования определениями на входном языке и средствами ассемблера.
25. Параллельные алгоритмы

Критерии и шкала оценивания по оценочному средству доклад, сообщение

Шкала оценивания (интервал баллов)	Критерий оценивания
5	Доклад (сообщение) представлен(о) на высоком уровне (студент в полном объеме осветил рассматриваемую проблематику, привел аргументы в пользу своих суждений, владеет профильным понятийным (категориальным) аппаратом и т.п.)
4	Доклад (сообщение) представлен(о) на среднем уровне (студент в целом осветил рассматриваемую проблематику, привел аргументы в пользу своих суждений, допустив некоторые неточности и т.п.)
3	Доклад (сообщение) представлен(о) на низком уровне (студент допустил существенные неточности, изложил материал с ошибками, не владеет в достаточной степени профильным категориальным аппаратом и т.п.)
2	Доклад (сообщение) представлен(о) на неудовлетворительном уровне или не представлен (студент не готов, не выполнил задание и т.п.)

Лабораторные работы

Цель работ

Изучение основных приемов модульной и эволюционной разработки программ с применением процедурной и объектно-ориентированной парадигм программирования. Анализ взаимосвязи между артефактами (программными объектами, разработанными программистом). Описание зависимостей между артефактами.

Порядок выполнения

1. Ознакомиться с описанием лабораторной работы.
2. Получить вариант задания у преподавателя.
3. Разработать процедурную и объектно-ориентированную программы в соответствии с условием задания.
4. Провести отладку и тестирование разработанных программ на одинаковых, заранее подготовленных наборах данных. Количество тестовых наборов данных – не менее пяти. Число уникальных элементов в тестовых наборах должно варьироваться от нуля до 100 и более. При необходимости, программа должна правильно обрабатывать переполнение по излишним данным.
5. Описать зависимость между артефактами процедурной и объектно-ориентированной программ.
6. Зафиксировать для отчета основные характеристики процедурной и объектно-ориентированной программ, такие как: общее количество модулей, число интерфейсных модулей и модулей реализации, количество артефактов, общий размер исходных текстов.
7. Провести сравнительный анализ процедурной и объектно-ориентированной программ по полученным характеристикам.

Содержание отчета

Полный отчет предоставляется в электронном виде, целиком размещаясь в специально выделенном каталоге. Он должен содержать:

1. Пояснительную записку, содержащую:
 - a) описание полученного задания;
 - b) описание зависимостей между артефактами и модулями процедурной и объектно-ориентированной программ;
 - c) основные характеристики процедурной и объектно-ориентированной программ, разработанных в ходе выполнения лабораторной работы;
 - d) выводы, полученные в ходе сравнительного анализа процедурной и объектно-ориентированной программ.

2. Исходные тексты процедурной и объектно-ориентированной программ, каждая из которых должна находиться в своем подкаталоге.
3. Тестовые наборы данных.
4. Результаты работы процедурной и объектно-ориентированной программ для различных тестовых наборов.

Темы лабораторных работ

ЛАБОРАТОРНАЯ РАБОТА №1

Тема: Разработка и анализ процедурной программы.

Разработка и анализ объектно-ориентированной программы.

ЛАБОРАТОРНАЯ РАБОТА №2

Тема: Добавление в процедурную и объектно-ориентированную программы дополнительных программных объектов, расширяющих номенклатуру обрабатываемых данных (абстрактных типов или классов).

ЛАБОРАТОРНАЯ РАБОТА №3

Тема: Добавление в процедурную и объектно-ориентированную программы дополнительных процедур для обработки данных (абстрактных типов или классов).

ЛАБОРАТОРНАЯ РАБОТА №4

Тема: Добавление полей для хранения дополнительных данных. Изменение функции ввода-вывода разработанных программных объектов с учетом проведенных добавлений.

ЛАБОРАТОРНАЯ РАБОТА №5

Тема: Добавление дополнительной процедуры, обеспечивающей решение требуемой задачи. Дополнительная процедура (метод) должна выполнять следующие функции: Сортировка элементов контейнера по заданному ключу.

ЛАБОРАТОРНАЯ РАБОТА №6

Тема: Добавление дополнительной процедуры, реализующей выборочный вывод из контейнера в файл сведений только о первом из двух видов объектов.

ЛАБОРАТОРНАЯ РАБОТА №7

Тема: Добавление дополнительной процедуры, реализующей мультиметод с двумя аргументами.

ЛАБОРАТОРНАЯ РАБОТА №8

Тема: Добавление в программы дополнительных программных объектов, расширяющие номенклатуру обрабатываемых данных (абстрактных типов или классов).

Варианты лабораторных работ

В ходе выполнения лабораторной работы необходимо написать процедурную и объектно-ориентированную программы, которые должны быть оформлены в виде консольных приложений, удовлетворяющих следующим требованиям:

1. Запуск программы осуществляется из командной строки, в которой указываются: имя запускаемой программы; имя файла с исходными данными; имя файла с выходными данными.
2. Для каждого программного объекта, загружаемого в контейнер, исходный файл должен содержать, признак, а также список необходимых параметров. Этот список должен быть представлен в формате, удобном для обработки компьютером.
3. Функция, формирующая выходной файл, должна выдавать в него программные объекты и их текущие параметры. Помимо этого необходимо вывести информацию об общем количестве объектов, содержащихся в контейнере. Информация должна быть представлена в форме, удобной для восприятия пользователем.
4. Программа должна иметь модульную структуру, соответствующую выданному варианту задания.
5. Процедурная программа должна иметь организацию обобщений, соответствующую выданному варианту задания.

Выбор варианта задания

Каждый из вариантов собирается из четырех независимых компонент: условия задачи, типа контейнера, вида модульной структуры программы, способа построения обобщения в процедурной программе. Выбор этих составляющих осуществляется на основе номера варианта задания (от 1 до 648), выданного преподавателем.

Пусть N_{var} – номер варианта задания, div – операция целочисленного деления, mod – операция выделения остатка от целочисленного деления. Тогда номер условия задачи N_{task} вычисляется следующим образом:

$$N_{\text{task}} = (((N_{\text{var}} - 1) \text{ div } 3) \text{ div } 3) \text{ div } 6 \text{ mod } 12 + 1.$$

Номер контейнера N_{cont} определяется по формуле:

$$N_{\text{cont}} = (((N_{\text{var}} - 1) \text{ div } 3) \text{ div } 3) \text{ mod } 6 + 1.$$

Номер модульной структуры N_{mod} определяется как:

$$N_{\text{mod}} = ((N_{\text{var}} - 1) \text{ div } 3) \text{ mod } 3 + 1.$$

Номер обобщения N_{union} вычисляется следующим образом:

$$N_{\text{union}} = (N_{\text{var}} - 1) \bmod 3 + 1.$$

Критерии и шкала оценивания по оценочному средству лабораторная работа

Шкала оценивания (интервал баллов)	Критерий оценивания
5	Лабораторная работа выполнена на высоком уровне (правильность выполнения 90-100%)
4	Лабораторная работа выполнена на среднем уровне (правильность выполнения 75-89%)
3	Лабораторная работа выполнена на низком уровне (правильность выполнения 50-74%)
2	Лабораторная работа выполнена на неудовлетворительном уровне (правильность выполнения менее чем на 50%)

Вопросы к контрольным работам:

1. В чем заключается основной метод обработки программы при отладке?
2. Что показывает результативную активность программы?
3. Когда принимается решение о продолжении незавершённых вычислений?
4. В каких пределах планируется функционирование участков повторяемости?
5. Каким способом гарантирована корректность сложной информационной обработки?
6. С какой целью производится оценка эффективности алгоритмов?
7. Какие характеристики используются при оценке эффективности алгоритмов?
8. Что выражает временная эффективность алгоритма? Поясните на примере.
9. Что выражает асимптотическая зависимость количества базовых операций от размера входных данных? Поясните на примере.
10. Что выражает амортизированная эффективность алгоритма? Поясните на примере.
11. Как провести оценку правильности алгоритма?
12. Что такое алфавит?
13. Что такое язык?
14. Что такое грамматика?
15. Что такое машинно-ориентированные языки программирования?
16. Что такое процедурно-ориентированные языки программирования?
17. Что происходит в процессе постановки задачи?

18. Что происходит в процессе составления программы?
19. Что происходит в процессе отладки программы?
20. Что происходит в процессе анализа результатов решения?
21. Что происходит в процессе переноса программы на машинный носитель?

Критерии и шкала оценивания по оценочному средству контрольная работа

Шкала оценивания (интервал баллов)	Критерий оценивания
5	Контрольная работа выполнена на высоком уровне (правильные ответы даны на 90-100% вопросов/задач)
4	Контрольная работа выполнена на среднем уровне (правильные ответы даны на 75-89% вопросов/задач)
3	Контрольная работа выполнена на низком уровне (правильные ответы даны на 50-74% вопросов/задач)
2	Контрольная работа выполнена на неудовлетворительном уровне (правильные ответы даны менее чем на 50%)

Оценочные средства для промежуточной аттестации (зачет)

1. Понятие "парадигма программирования" и его история. Взаимосвязь языков программирования и парадигм программирования.
2. Рекурсия. Виды рекурсии. Примеры.
3. Остаточная (хвостовая) рекурсия.
4. Императивное и императивно-процедурное программирование: ключевые понятия.
5. Язык Лисп. Понятие S-выражения.
6. Язык Лисп. Функции высоких порядков, мапперы.
7. Язык Лисп. Лексическое и динамическое связывание. Лексические замыкания.
8. Функция REDUCE и примеры её использования.
9. Понятие функционального программирования. Ключевые особенности и термины.
10. Основные понятия языка Пролог: атом, терм, предикат, переменная. Сопоставление (унификация).
11. Выполнение программы на Прологе. Декларативная и процедурная семантика. Отрицание и отсечение.
12. Инвертирование предикатов в Прологе. Причины, по которым инвертирование может не работать.
13. Логическое программирование и базы данных. Понятие декларативного программирования.

14. Основы языка Рефал. Ситуационное программирование. Поиск с возвратами в Рефале-5.
15. Понятие ленивых вычислений. Достоинства и недостатки.
16. Основы языка Норе.
17. Обработка потенциально бесконечных структур данных с помощью ленивых вычислений.
18. Командно-скриптовые языки программирования и их ключевые особенности.
19. "Текст" как универсальное представление.
20. Основы объектно-ориентированного программирования. Понятие класса как множества.
21. Наследование в ООП. Наследование как сужение множества. Виртуальные методы.
22. Полиморфизм, его теоретико-множественное и техническое описание.
23. Обобщенное программирование.
24. Обработка исключительных ситуаций и её связь с основными стилями программирования.
25. Событийно-ориентированное программирование.
26. Основные методы многостилевой (мультипарадигмальной) интеграции.

Критерии и шкала оценивания по оценочному средству промежуточный контроль (зачет)

Характеристика знания предмета и ответов	Зачеты
Студент глубоко и в полном объёме владеет программным материалом. Грамотно, исчерпывающе и логично его излагает в устной или письменной форме. При этом знает рекомендованную литературу, проявляет творческий подход в ответах на вопросы и правильно обосновывает принятые решения, хорошо владеет умениями и навыками при выполнении практических задач.	зачтено
Студент знает программный материал, грамотно и по сути излагает его в устной или письменной форме, допуская незначительные неточности в утверждениях, трактовках, определениях и категориях или незначительное количество ошибок. При этом владеет необходимыми умениями и навыками при выполнении практических задач.	
Студент знает только основной программный материал, допускает неточности, недостаточно чёткие формулировки, непоследовательность в ответах, излагаемых в устной или письменной форме. При этом недостаточно владеет умениями и навыками при выполнении практических задач. Допускает до 30% ошибок в излагаемых ответах.	
Студент не знает значительной части программного материала. При этом допускает принципиальные ошибки в доказательствах, в трактовке понятий и категорий, проявляет низкую культуру знаний, не владеет основными умениями и навыками при выполнении практических задач. Студент отказывается от ответов на дополнительные вопросы.	не зачтено

Лист изменений и дополнений

№ п/п	Виды дополнений и изменений	Дата и номер протокола заседания кафедры (кафедр), на котором были рассмотрены и одобрены изменения и дополнения	Подпись (с расшифровкой) заведующего кафедрой (заведующих кафедрами)

Экспертное заключение

Представленный фонд оценочных средств (далее - ФОС) *по дисциплине «Современные парадигмы программирования»* соответствует требованиям ФГОС ВО.

Предлагаемые формы и средства текущего и промежуточного контроля адекватны целям и задачам реализации основной образовательной программы по направлению подготовки 09.03.03 Прикладная информатика.

Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины и учебно-методическое обеспечение самостоятельной работы обучающегося представлены в полном объеме.

Виды оценочных средств, включенные в представленный фонд, отвечают основным принципам формирования ФОС.

Разработанный и представленный для экспертизы фонд оценочных средств рекомендуется к использованию в процессе подготовки бакалавров, по указанному направлению.

Председатель учебно-
методической комиссии
факультета компьютерных систем
и информационных технологий



Ветрова Н. Н.