



Федеральное агентство морского и речного транспорта
Федеральное государственное бюджетное образовательное учреждение
высшего образования

**«Государственный университет морского и речного флота
имени адмирала С.О. Макарова»**

Воронежский филиал ФГБОУ ВО «ГУМРФ имени адмирала С.О. Макарова»

Кафедра математики, информационных систем и технологий

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

по дисциплине *«Синтез программных систем»*
(приложение к рабочей программе дисциплины)

Направление подготовки 09.03.02 Информационные системы и технологии

Направленность (профиль) Информационные системы на транспорте

Уровень высшего образования бакалавриат

Форма обучения заочная

Воронеж
2023

1. Перечень компетенций и этапы их формирования в процессе освоения дисциплины

Рабочей программой дисциплины «Синтез программных систем» предусмотрено формирование следующих компетенций

Таблица 1

Перечень компетенций и этапы их формирования в процессе освоения дисциплины

Код и наименование компетенции	Код индикатора достижения компетенции	Планируемые результаты обучения по дисциплине
УК-3 Способен осуществлять социальное взаимодействие и реализовывать свою роль в команде	УК-3.4 Осуществление обмена информацией, знаниями и опытом с членами команды, соблюдение установленных норм и правил командной работы	Знать: методы коммуникации и контроля при разработке программного обеспечения Уметь: выстраивать план коммуникаций и взаимодействий с командой проекта по разработке ПО Владеть: навыками обмена информацией, знаниями и опытом с членами команды, соблюдения установленных норм и правил командной работы
ПК-2 Способен осуществлять техническую поддержку и создавать техническую документацию на продукцию в сфере информационных технологий	ПК-2.2 Создание технической документации на продукцию в сфере информационных технологий	Знать: процессы разработки ПО, основы проектирования ПО Уметь: формировать и анализировать требования к ПО Владеть: навыками создания технической документации на продукцию в сфере информационных технологий

2. Паспорт фонда оценочных средств для проведения текущей и промежуточной аттестации обучающихся

Таблица 2

Оценочные средства для проведения текущей и промежуточной аттестации обучающихся

№ п/п	Наименование раздела (темы) дисциплины	Формируемая компетенция	Наименование оценочного средства
1	Организация процесса разработки	ПК-2	Тестирование, лабораторные работы, РГР, экзамен
2	Руководство программным проектом	УК-3	Тестирование, лабораторные работы, РГР, экзамен
3	Оценка при планировании программного проекта	ПК-2	Тестирование, лабораторные

№ п/п	Наименование раздела (темы) дисциплины	Формируемая компетенция	Наименование оценочного средства
			<i>работы, РГР, экзамен</i>
4	Формирование и анализ требований	УК-3 ПК-2	<i>Тестирование, лабораторные работы, РГР, экзамен</i>
5	Основы проектирования программных систем	ПК-2	<i>Тестирование, лабораторные работы, РГР, экзамен</i>
6	Классические методы проектирования	ПК-2	<i>Тестирование, лабораторные работы, РГР, экзамен</i>

Таблица 3

Критерии оценивания результата обучения по дисциплине и шкала оценивания по дисциплине

Результат обучения по дисциплине	Критерии оценивания результата обучения по дисциплине и шкала оценивания по дисциплине				Процедура оценивания
	2	3	4	5	
	Не зачтено	Зачтено			
УК-3.1 Знать: методы коммуникации и контроля при разработке программного обеспечения	<i>Отсутствие или фрагментарные представления о методах коммуникации и контроля при разработке программного обеспечения</i>	<i>Неполные представления о методах коммуникации и контроля при разработке программного обеспечения</i>	<i>Сформированные, но содержащие отдельные пробелы представления о методах коммуникации и контроля при разработке программного обеспечения.</i>	<i>Сформированные систематические представления о методах коммуникации и контроля при разработке программного обеспечения</i>	<i>Тестирование, лабораторные работы, РГР, экзамен</i>
УК-3. 1 Уметь: выстраивать план коммуникаций и взаимодействий с командой проекта по разработке ПО	<i>Отсутствие умений или фрагментарные умения выстраивать план коммуникаций и взаимодействий с командой проекта по разработке ПО</i>	<i>В целом удовлетворительные, но не систематизированные умения выстраивать план коммуникаций и взаимодействий с командой проекта по разработке ПО</i>	<i>В целом удовлетворительные, но содержащие отдельные пробелы умения выстраивать план коммуникаций и взаимодействий с командой проекта по разработке ПО</i>	<i>Сформированные умения выстраивать план коммуникаций и взаимодействий с командой проекта по разработке ПО</i>	<i>Тестирование, лабораторные работы, РГР, экзамен</i>
УК-3.1 Владеть: навыками обмена информацией, знаниями и опытом с членами команды, соблюдения установленных норм и правил командной работы	<i>Отсутствие владения или фрагментарное владение навыками обмена информацией, знаниями и опытом с членами</i>	<i>В целом удовлетворительные, но не систематизированные навыки обмена информацией, знаниями и опытом с членами команды,</i>	<i>В целом удовлетворительные, но содержащие отдельные пробелы навыки обмена информацией, знаниями и опытом с членами</i>	<i>Сформированное владение навыками обмена информацией, знаниями и опытом с членами команды, соблюдения</i>	<i>Тестирование, лабораторные работы, РГР, экзамен</i>

	<i>команды, соблюдения установленных норм и правил командной работы</i>	<i>соблюдения установленных норм и правил командной работы</i>	<i>команды, соблюдения установленных норм и правил командной работы</i>	<i>установленных норм и правил командной работы</i>	
ПК-2. 2 Знать: процессы разработки ПО, основы проектирования ПО	<i>Отсутствие или фрагментарные представления о процессах разработки ПО, основах проектирования ПО</i>	<i>Неполные представления о процессах разработки ПО, основах проектирования ПО</i>	<i>Сформированные, но содержащие отдельные пробелы представления о процессах разработки ПО, основах проектирования ПО.</i>	<i>Сформированные систематические представления о процессах разработки ПО, основах проектирования ПО</i>	<i>Тестирование, лабораторные работы, РГР, экзамен</i>
ПК-2.2 Уметь: формировать и анализировать требования к ПО	<i>Отсутствие умений или фрагментарные умения формировать и анализировать требования к ПО</i>	<i>В целом удовлетворительные, но не систематизированные умения формировать и анализировать требования к ПО</i>	<i>В целом удовлетворительные, но содержащие отдельные пробелы умения формировать и анализировать требования к ПО</i>	<i>Сформированные умения формировать и анализировать требования к ПО</i>	<i>Тестирование, лабораторные работы, РГР, экзамен</i>
ПК-2.2 Владеть: навыками создания технической документации на продукцию в сфере информационных технологий	<i>Отсутствие владения или фрагментарное владение навыками создания технической документации на продукцию в сфере информационных технологий</i>	<i>В целом удовлетворительные, но не систематизированные навыки создания технической документации на продукцию в сфере информационных технологий</i>	<i>В целом удовлетворительные, но содержащие отдельные пробелы навыки создания технической документации на продукцию в сфере информационных технологий</i>	<i>Сформированное владение навыками создания технической документации и на продукцию в сфере информационных технологий</i>	<i>Тестирование, лабораторные работы, РГР, экзамен</i>

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ ТЕКУЩЕГО КОНТРОЛЯ

Тесты текущего контроля

Тест текущего контроля 1

Выполните тест. В каждом тестовом вопросе – один верный вариант ответа.

1. В стадии разработки программы не входит:
 - a) автоматизация программирования;
 - b) постановка задачи;
 - c) составление спецификаций;
 - d) эскизный проект;
 - e) тестирование.

2. Самый важный критерий качества программы:
 - a) работоспособность;
 - b) надежность;
 - c) эффективность;
 - d) быстродействие;
 - e) простота эксплуатации.

3. Способы оценки качества:
 - a) сравнение с аналогами;
 - b) наличие документации;
 - c) оптимизация программы;
 - d) структурирование алгоритма.

4. Существует ли связь между эффективностью и оптимизацией программы:
 - a) да;
 - b) нет.

5. Наиболее важный критерий качества:
 - a) надежность;
 - b) быстродействие;
 - c) удобство в эксплуатации;
 - d) удобный интерфейс;
 - e) эффективность.

6. Способы оценки надежности:
 - a) тестирование;
 - b) сравнение с аналогами;
 - c) трассировка;
 - d) оптимизация.

7. Повышает ли качество программ оптимизация:
 - a) да;
 - b) нет.

8. Существует ли связь между надежностью и быстродействием:
 - a) нет;
 - b) да.

9. В каких единицах можно измерить надежность:
 - a) отказов/час;
 - b) км/час;
 - c) Кбайт/сек;
 - d) операций/сек.

10. В каких единицах можно измерить быстродействие:
 - a) отказов/час;

- b) км/час;
- c) Кбайт/сек;
- d) операций/сек.

11. Нахождение наилучшего варианта из множества возможных:

- a) оптимизация;
- b) тестирование;
- c) автоматизация;
- d) отладка;
- e) сопровождение.

12. Что такое оптимизация программ:

- a) улучшение работы существующей программы;
- b) создание удобного интерфейса пользователя;
- c) разработка модульной конструкции программы;
- d) применение методов объектно-ориентированного программирования.

13. Критерии оптимизации:

- a) время выполнения или размер требуемой памяти;
- b) размер программы и ее эффективность;
- c) независимость модулей;
- d) качество программы, ее надежность.

14. Критерии оптимизации:

- a) эффективность использования ресурсов;
- b) структурирование алгоритма;
- c) структурирование программы.

15. Возможна ли оптимизация программ без участия программиста:

- a) да;
- b) нет.

16. Возможна ли оптимизация циклов:

- a) да;
- b) нет.

17. В чем заключается оптимизация условных выражений:

- a) в изменении порядка следования элементов выражения;
- b) в использовании простых логических выражений;
- c) в использовании сложных логических выражений;
- d) в использовании операций AND, OR и NOT.

18. Оптимизация циклов заключается в:

- a) уменьшении количества повторений тела цикла;
- b) просмотре задачи с другой стороны;
- c) упрощение задачи за счет включения логических операций.

19. Оптимизация программы это:

- a) модификация;
- b) отладка;
- c) повышение сложности программы;
- d) уменьшение сложности программы.

20. Критерии оптимизации программы:

- a) быстроедействие или размер программы;
- b) быстроедействие и размер программы;
- c) надежность или эффективность;
- d) надежность и эффективность.

21. Результат оптимизации программы:

- a) эффективность;
- b) надежность;
- c) машино-независимость;
- d) мобильность.

22. Сущность оптимизации циклов:

- a) сокращение количества повторений выполнения тела цикла;
- b) сокращение тела цикла;
- c) представление циклов в виде блок-схем;
- d) трассировка циклов;
- e) поиск ошибок в циклах.

Тест текущего контроля 2

Выполните тест. В каждом тестовом вопросе – один верный вариант ответа.

1. Что относится к этапу программирования:

- a) написание кода программы;
- b) разработка интерфейса;
- c) работоспособность;
- d) анализ требований.

2. Последовательность этапов программирования:

- a) компилирование, компоновка, отладка;
- b) компоновка, отладка, компилирование;
- c) отладка, компилирование, компоновка;
- d) компилирование, отладка, компоновка.

3. Инструментальные средства программирования:

- a) компиляторы, интерпретаторы;
- b) СУБД (системы управления базами данных);
- c) BIOS (базовая система ввода-вывода);
- d) ОС (операционные системы).

4. На языке программирования составляется:
- a) исходный код;
 - b) исполняемый код;
 - c) объектный код;
 - d) алгоритм.
5. Правила, которым должна следовать программа это:
- a) алгоритм;
 - b) структура;
 - c) спецификация;
 - d) состав информации.
6. Можно ли внутри цикла поместить еще один цикл:
- a) да;
 - b) нет.
7. Можно ли внутри условного оператора поместить еще одно условие:
- a) да;
 - b) нет.
8. Можно ли одно большое (длинное) выражение разбить на несколько выражений:
- a) да;
 - b) нет.
9. Если имеется стандартная функция, нужно ли писать собственную:
- a) нет;
 - b) да.
10. Доступ, при котором записи файла читаются в физической последовательности, называется:
- a) последовательным;
 - b) прямым;
 - c) простым;
 - d) основным.
11. Доступ, при котором записи файла обрабатываются в произвольной последовательности, называется:
- a) прямым;
 - b) последовательным;
 - c) простым;
 - d) основным.
12. Методы программирования (укажите НЕ верный ответ):
- a) логическое;

- b) структурное;
- c) модульное.

13. Что выполняется раньше:

- a) разработка алгоритма;
- b) выбор языка программирования;
- c) написание исходного кода;
- d) компиляция.

Тест текущего контроля 3

Выполните тест. В каждом тестовом вопросе – один верный вариант ответа.

1. В чем сущность модульного программирования:

- a) в разбиении программы на отдельные функционально независимые части;
- b) в разбиении программы на отдельные равные части;
- c) в разбиение программы на процедуры и функции;

2. Можно ли сочетать модульное и структурное программирование:

- a) да;
- b) нет.

3. Может ли модуль включать несколько процедур или функций:

- a) да;
- b) нет.

4. Рекомендуемые размеры модулей:

- a) небольшие;
- b) большие;
- c) равные;
- d) фиксированной длины.

5. В чем заключается независимость модуля:

- a) в написании, отладке и тестировании независимо от остальных модулей;
- b) в разработке и написании независимо от других модулей;
- c) в независимости от работы основной программы.

6. При модульном программировании желательно, чтобы модуль имел:

- a) большой размер;
- b) небольшой размер;
- c) фиксированный размер;
- d) любой размер.

7. Модульное программирование это:

- a) разбиение программы на отдельные части;

- b) структурирование;
- c) использование стандартных процедур и функций.

8. Можно ли использовать оператор GO TO в модульных программах:

- a) можно;
- b) нельзя.

9. Разрешается ли использование циклов при модульном программировании:

- a) да;
- b) нет.

10. Разрешается ли использование условных операторов при модульном программировании:

- a) да;
- b) нет.

11. Сократится ли размер программы, если ее написать в виде набора модулей:

- a) нет;
- b) да.

12. Достоинство модульного программирования:

- a) создание программы по частям в произвольном порядке;
- b) не требует компоновки;
- c) всегда дает эффективные программы;
- d) снижает количество ошибок.

137. Недостаток модульного программирования:

- a) увеличивает трудоемкость программирования;
- b) усложняет процедуру комплексного тестирования;
- c) снижает быстродействие программы;
- d) не позволяет выполнять оптимизацию программы.

13. Достоинство модульного программирования:

- a) возможность приступить к тестированию до завершения написания всей программы;
- b) не требует комплексного тестирования;
- c) уменьшает размер программы;
- d) повышает надежность программы.

14. Допустимо ли использование оператора GO TO при структурном программировании:

- a) нет;
- b) да.

15. Можно ли сочетать структурное программирование с модульным:

- a) можно;

- b) нельзя;
- c) только в особых случаях.

16. Любую ли программу можно привести к структурированному виду:

- a) любую;
- b) не все;
- c) нельзя.

17. Можно ли использовать оператор GO TO в структурированных программах:

- a) можно;
- b) нельзя;
- c) только в особых случаях.

18. Возможно, ли преобразовать неструктурированную программу к структурному виду:

- a) да;
- b) нет.

19. Возможно ли программирование без оператора GO TO:

- a) да;
- b) нет.

20. При структурном программировании задача выполняется:

- a) поэтапным разбиением на более легкие задачи;
- b) без участия программиста;
- c) объединением отдельных модулей программы.

21. Разрешается ли использование оператора GO TO при структурном программировании:

- a) нет;
- b) да;
- c) иногда.

22. Разрешается ли использование циклов при структурном программировании:

- a) да;
- b) нет.

23. Разрешается ли использование оператора IF при структурном программировании:

- a) да;
- b) нет.

24. Программирование без GO TO применяется. при:

- a) структурном программировании;
- b) модульном программировании;

- c) объектно-ориентированном программировании;
- d) все ответы верные.

25. Достоинство структурного программирования:

- a) можно приступить к комплексному тестированию на раннем этапе разработки;
- b) можно приступить к автономному тестированию на раннем этапе разработки;
- c) нет необходимости выполнять тестирование;
- d) можно пренебречь отладкой.

Тест текущего контроля 4

Выполните тест. В каждом тестовом вопросе – один верный вариант ответа.

1. Что относится к этапу программирования:

- a) написание кода программы;
- b) разработка интерфейса;
- c) работоспособность;
- d) анализ требований.

2. Последовательность этапов программирования:

- a) компилирование, компоновка, отладка;
- b) компоновка, отладка, компилирование;
- c) отладка, компилирование, компоновка;
- d) компилирование, отладка, компоновка.

3. Инструментальные средства программирования:

- a) компиляторы, интерпретаторы;
- b) СУБД (системы управления базами данных);
- c) BIOS (базовая система ввода-вывода);
- d) ОС (операционные системы).

4. На языке программирования составляется:

- a) исходный код;
- b) исполняемый код;
- c) объектный код;
- d) алгоритм.

5. Правила, которым должна следовать программа это:

- a) алгоритм;
- b) структура;
- c) спецификация;
- d) состав информации.

6. Можно ли внутри цикла поместить еще один цикл:

- a) да;
- b) нет.

7. Можно ли внутри условного оператора поместить еще одно условие:

- a) да;
- b) нет.

8. Можно ли одно большое (длинное) выражение разбить на несколько выражений:

- a) да;
- b) нет.

9. Если имеется стандартная функция, нужно ли писать собственную:

- c) нет;
- d) да.

10. Доступ, при котором записи файла читаются в физической последовательности, называется:

- a) последовательным;
- b) прямым;
- c) простым;
- d) основным.

11. Доступ, при котором записи файла обрабатываются в произвольной последовательности, называется:

- a) прямым;
- b) последовательным;
- c) простым;
- d) основным.

12. Методы программирования (укажите НЕ верный ответ):

- a) логическое;
- b) структурное;
- c) модульное.

13. Что выполняется раньше:

- a) разработка алгоритма;
- b) выбор языка программирования;
- c) написание исходного кода;
- d) компиляция.

14. Можно ли переменным присваивать произвольные идентификаторы:

- a) да;
- b) нет.

15. Найдите НЕ правильное условие для создания имен:

- a) имена могут содержать пробелы;
- b) длинное имя можно сократить;
- c) из имени лучше выбрасывать гласные;
- d) можно использовать большие буквы.

16. Какие символы не допускаются в именах переменных:

- a) пробелы;
- b) цифры;
- c) подчеркивание

17. Можно ли использовать имена, которые уже были использованы в другой программе (модуле):

- a) да;
- b) нет.

18. Можно ли ставить знак подчеркивания в начале имени:

- a) да, но не рекомендуется;
- b) да, без ограничений;
- c) нет.

19. Как называется способ составления имен переменных, когда в начале имени сообщается тип переменной:

- a) прямым указанием;
- b) венгерской нотацией;
- c) структурным программированием;
- d) поляризацией.

20. Можно ли писать комментарии в отдельной строке:

- a) да;
- b) нет.

21. Транслируются ли комментарии:

- a) да;
- b) нет.

22. Наличие комментариев позволяет:

- a) быстрее найти ошибки в программе;
- b) быстрее писать программы;
- c) быстрее выполнять программы.

23. Наличие комментариев позволяет:

- a) легче разобраться в программе;
- b) применять сложные структуры;
- c) увеличить быстродействие.

24. Наличие комментариев позволяет:
- a) улучшить читабельность программы;
 - b) улучшить эксплуатацию программы;
 - c) повысить надежность программы.
25. Что определяет выбор языка программирования:
- a) область приложения;
 - b) знание языка;
 - c) наличие дополнительных библиотек.
26. Возможно ли комбинирование языков программирования в рамках одной задачи:
- a) да;
 - b) нет.
27. Допустимо ли комбинирование языков программирования в рамках одной задачи :
- a) да;
 - b) нет.
28. Для каких задач характерно использование большого количества исходных данных, выполнение операций поиска, группировки:
- a) для экономических задач;
 - b) для системных задач;
 - c) для инженерных задач.
29. Для каких задач характерен большой объем вычислений, использование сложного математического аппарата:
- a) для инженерных задач;
 - b) для системных задач;
 - c) для экономических задач.
30. На каком этапе производится выбор языка программирования:
- a) проектирование;
 - b) программирование;
 - c) отладка;
 - d) тестирование.
31. Можно ли использовать комбинацию языков программирования в рамках одного проекта:
- a) да;
 - b) нет.
32. Для решения экономических задач характерно применение:
- a) СУБД (систем управления базами данных);
 - b) языков высокого уровня;

- c) языков низкого уровня;
- d) применение сложных математических расчетов.

33. Для решения инженерных задач характерно применение:

- a) САПР (систем автоматизированного проектирования);
- b) СУБД (систем управления базами данных);
- c) ОС (операционных систем).

Тест текущего контроля 5

Выполните тест. В каждом тестовом вопросе – один верный вариант ответа.

1. Что такое автоматизация программирования:

- a) создание исходного кода программными средствами;
- b) создание исходного кода при помощи компилятора;
- c) создание исходного кода без разработки алгоритма.

2. В чем сущность автоматизации программирования:

- a) создание программы без написания ее текста;
- b) получение готовой программы без выполнения компоновки;
- c) в отсутствии компиляции.

3. Возможна ли автоматизация программирования:

- a) да;
- b) нет.

4. Создание исполняемого кода программы без написания исходного кода называется:

- a) составлением спецификаций;
- b) отладкой;
- c) проектированием.
- d) автоматизацией программирования;

5. Одно из преимуществ автоматизации программирования:

- a) наглядное программирование с визуальным контролем;
- b) получение стандартной программы;
- c) создание программы с оптимальным кодом.

6. Один из методов автоматизации программирования:

- a) структурное программирование;
- b) модульное программирование;
- c) визуальное программирование;
- d) объектно-ориентированное программирование.

7. Влияет ли автоматизация программирования на эффективность программы:

- a) нет;
- b) да

8. Автоматизация программирования позволяет:
- a) повысить надежность программы;
 - b) сократить время разработки программы;
 - c) повысить быстродействие программы.
9. Позволяет ли автоматизация программирования всегда создавать эффективные программы:
- a) да.
 - b) нет;
10. Позволяет ли автоматизация программирования всегда создавать надежные программы:
- a) нет;
 - b) да.
11. Недостаток автоматизации программирования;
- a) низкое быстродействие;
 - b) большой размер программы;
 - c) сложность программы.
12. Возможны ли ошибки при автоматизации программирования:
- a) да;
 - b) нет.
13. Нужно ли выполнять тестирование при автоматизации программирования:
- a) да;
 - b) нет.
14. Выполняется ли процедура компиляции при автоматизации программирования:
- a) да;
 - b) нет.
15. Что легко поддается автоматизации:
- a) интерфейс;
 - b) работа с файлами;
 - c) сложные логические задачи;
 - d) алгоритмизация.
16. Относится ли визуальное программирование к средствам автоматизации:
- a) да;
 - b) нет.

Тест текущего контроля 6

Выполните тест. В каждом тестовом вопросе – один верный вариант ответа.

1. Причины синтаксических ошибок:

- a) плохое знание языка программирования;
- b) ошибки в исходных данных;
- c) ошибки, допущенные на более ранних этапах;
- d) неправильное применение процедуры тестирования.

2. Когда можно обнаружить синтаксические ошибки:

- a) при компиляции;
- b) при отладке;
- c) при тестировании;
- d) на этапе проектирования;
- e) при эксплуатации.

3. Ошибки компоновки заключаются в том, что:

- a) указано внешнее имя, но не объявлено;
- b) неправильно использовано зарезервированное слово;
- c) составлено неверное выражение;
- d) указан неверный тип переменной.

4. Могут ли проявиться ошибки при изменении условий эксплуатации:

- a) да;
- b) нет.

5. Могут ли проявиться ошибки при изменении в предметной области:

- a) да;
- b) нет.

6. Возможно ли программирование с защитой от ошибок:

- a) да;
- b) нет.

7. Есть ли недостатки программирования с защитой от ошибок:

- a) да;
- b) нет.

8. Защитное программирование это:

- a) встраивание в программу отладочных средств;
- b) создание задач защищенных от копирования;
- c) разделение доступа в программе;
- d) использование паролей;
- e) оформление авторских прав на программу.

9. Вид ошибки с неправильным написанием служебных слов (операторов):

- a) синтаксическая;
- b) семантическая;
- c) логическая;
- d) символная.

10. Вид ошибки с неправильным использованием служебных слов (операторов):

- a) семантическая;
- b) синтаксическая;
- c) логическая;
- d) символная.

11. Ошибки при написании программы бывают:

- a) синтаксические;
- b) орфографические;
- c) лексические;
- d) фонетические;
- e) морфологические.

12. Процедура поиска ошибки, когда известно, что она есть это:

- a) отладка;
- b) тестирование;
- c) компоновка;
- d) транзакция;
- e) трансляция.

13. Программа для просмотра значений переменных при выполнении программы:

- a) отладчик;
- b) компилятор;
- c) интерпретатор;
- d) трассировка;
- e) тестирование.

14. Отладка – это:

- a) процедура поиска ошибок, когда известно, что ошибка есть;
- b) определение списка параметров;
- c) правило вызова процедур (функций);
- d) составление блок-схемы алгоритма.

15. Когда программист может проследить последовательность выполнения команд программы:

- a) при трассировке;
- b) при тестировании;
- c) при компиляции;
- d) при выполнении программы;
- e) при компоновке.

16. На каком этапе создания программы могут появиться синтаксические ошибки:

- a) программирование;
- b) проектирование;
- c) анализ требований;
- d) тестирование.

17. Когда приступают к тестированию программы:

- a) когда программа уже закончена;
- b) после постановки задачи;
- c) на этапе программирования;
- d) на этапе проектирования;
- e) после составления спецификаций,

18. Тестирование бывает:

- a) автономное;
- b) инструментальное;
- c) визуальное;
- d) алгоритмическое.

19. Тестирование бывает:

- a) комплексное;
- b) инструментальное;
- c) визуальное;
- d) алгоритмическое.

20. Существует ли различие между отладкой и тестированием:

- a) да;
- b) нет.

21. При комплексном тестировании проверяются:

- a) согласованность работы отдельных частей программы;
- b) правильность работы отдельных частей программы;
- c) быстродействие программы;
- d) эффективность программы.

22. Чему нужно уделять больше времени, чтобы получить хорошую программу:

- a) тестированию;
- b) программированию;
- c) отладке;
- d) проектированию.

23. Процесс исполнения программы с целью обнаружения ошибок:

- a) тестирование;
- b) кодирование;
- c) сопровождение;

d) проектирование.

24. Автономное тестирование это:

- a) тестирование отдельных частей программы;
- b) инструментальное средство отладки;
- c) составление блок-схем;
- d) пошаговая проверка выполнения программы.

25. Трассировка это:

- a) проверка пошагового выполнения программы;
- b) тестирование исходного кода;
- c) отладка модуля;
- d) составление блок-схемы алгоритма.

26. Локализация ошибки:

- a) определение места возникновения ошибки;
- b) определение причин ошибки;
- c) обнаружение причин ошибки;
- d) исправление ошибки.

27. Назначение тестирования:

- a) повышение надежности программы;
- b) обнаружение ошибок;
- c) повышение эффективности программы;
- d) улучшение эксплуатационных характеристик;
- e) приведение программы к структурированному виду.

28. Назначение отладки:

- a) поиск причин существующих ошибок;
- b) поиск возможных ошибок;
- c) составление спецификаций;
- d) разработка алгоритма.

29. Инструментальные средства отладки (НЕ правильный ответ):

- a) компиляторы;
- b) отладчики;
- c) трассировка.

30. Отладка программ это:

- a) локализация и исправление ошибок;
- b) алгоритмизация программирования;
- c) компиляция и компоновка.

31. Что выполняется раньше, автономная или комплексная отладка:

- a) автономная;
- b) комплексная.

32. Что выполняется раньше, отладка или тестирование:

- a) отладка;
- b) тестирование.

Критерии оценки результатов тестирования

Оценка результатов тестирования. За каждый правильный ответ начисляется 1 балл. Для перевода баллов в оценку применяется универсальная шкала оценки образовательных достижений. Если обучающийся набирает

- от 90 до 100% от максимально возможной суммы баллов - выставляется оценка «отлично»;
- от 80 до 89% - оценка «хорошо»,
- от 51 до 79% - оценка «удовлетворительно»,
- менее 51% - оценка «неудовлетворительно».

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОГО КОНТРОЛЯ

Промежуточная аттестация – экзамен.

Вопросы для проведения экзамена

1. Цели и задачи технологий разработки ПО. Особенности современных крупных проектов разработки ПО.
2. Понятие программная инженерия. Основные, вспомогательные и организационные процессы программной инженерии.
3. Структурный подход к проектированию ПО. Сущность структурного подхода.
4. Объектно-ориентированная разработка программ. Объектно-ориентированные языки программирования. Объектно-ориентированные методологии разработки программных систем.
5. Каскадная модель жизненного цикла ПС: содержание этапов, область применения, достоинства и недостатки.
6. Эволюционная модель жизненного цикла ПС: последовательность действий, область применения, достоинства и недостатки.
7. Спиральная модель разработки ПО: содержание этапов создания ПС, область применения, достоинства и недостатки.
8. Инкрементальная модель разработки ПО. Развитие инкрементального подхода. XP-процессы.
9. Понятие программного проекта. Управление программным проектом. План и содержание его разделов. Составление сетевого графика работ.
11. Состав и структура коллектива разработчиков программного продукт, их функции. Составление расписания (PERT-диаграммы)
12. Управление документацией разработки программного продукта.

13. Рациональный Унифицированный Процесс. Динамические аспекты процессов: структура ЖЦ, стадии, итерации и контрольные точки.
14. Рациональный Унифицированный Процесс. Статическое содержание процесса: виды деятельности (технологические операции), рабочие продукты, исполнители и дисциплины (технологические процессы).
15. Внешнее описание программного средства и спецификация. Виды требований к ПО: системные, функциональные, характеристики качества.
16. Методы определения и формализация требований к ПО.
17. Понятие качества ПО и его многоуровневая модель. Характеристики и атрибуты качества.
18. Разработка требований к ПО: формирование и анализ, документирование, аттестация. Управление.
19. Алгоритмическая декомпозиция. Модульное программирование. Характеристики программного модуля.
20. Модели архитектур с различными способами обмена данными: репозиторий, «клиент-сервер».
21. Архитектуры с различными моделями управления.
22. Событийно-управляемые архитектуры.
23. Модели архитектур с различными подходами к обработке данных: непрерывная обработка, каналы и фильтры.
24. Объектно-ориентированная декомпозиция. Общая характеристика объектов. Виды отношений между объектами. Агрегация.
25. Абстрагирование. Общая характеристика классов. Виды отношений между классами. Ассоциации классов. Наследование. Полиморфизм. Агрегация.
26. Повторное использование компонентов. Инкапсуляция. Интерфейсы. Компонентная объектная модель (СОМ).
27. Принципы проектирования пользовательского интерфейса.
28. Структурное тестирование. Покрытие операторов, ветвей, условий.
29. Функциональное тестирование. Метод эквивалентного разбиения, граничных значений, причинно-следственных (функциональных) диаграмм.
30. Тестирование интеграции компонентов ПО: нисходящее и восходящее. Понятие драйвер и заглушка. Стохастическое тестирование.
31. Разработка программной документации. С-документация и П-документация.
32. Отладка ПО: цели и методы.
33. Управление конфигурацией ПО. Системы контроля версий. Регрессионное тестирование.
34. Аттестация ПО. Оценка качества ПО.
35. Инструментальные средства разработки ПО. Автоматизация разработки ПО. CASE-средства.
36. Сопровождение ПО. Основные подходы: с целью исправления ошибок, адаптации и изменения функциональных возможностей. Решение проблемы эволюции ПО – рефакторинг, реинженерия, реверсная инженерия.

**Показатели, критерии и шкала оценивания
письменных ответов на экзамене**

Критерии оценивания	Показатели и шкала оценивания			
	5	4	3	2
текущая аттестация	выполнение требований по текущей аттестации в полном объеме		выполнение требований по текущей аттестации в неполном объеме	невыполнение требований по текущей аттестации
полнота и правильность ответа	обучающийся полно излагает материал, дает правильное определение основных понятий	обучающийся достаточно полно излагает материал, однако допускает 1-2 ошибки, которые сам же исправляет, и 1-2 недочета в последовательности и языковом оформлении излагаемого	обучающийся демонстрирует знание и понимание основных положений данной темы, но излагает материал неполно и допускает неточности в определении понятий или формулировке правил	обучающийся демонстрирует незнание большей части соответствующего вопроса
степень осознанности, понимания изученного	демонстрирует понимание материала, может обосновать свои суждения, применить знания на практике, привести необходимые примеры не только из учебника, но и самостоятельно составленные	присутствуют 1-2 недочета в обосновании своих суждений, количество приводимых примеров ограничено	не умеет достаточно глубоко и доказательно обосновать свои суждения и привести свои примеры	допускает ошибки в формулировке определений и правил, искажающие их смысл
языковое оформление ответа	излагает материал последовательно и правильно с точки зрения норм литературного	излагает материал последовательно, с 2-3 ошибками в языковом оформлении	излагает материал непоследовательно и допускает много ошибок в языковом оформлении излагаемого	беспорядочно и неуверенно излагает материал

	языка			
--	-------	--	--	--

Составитель: к.ф.-м.н., доцент Черняева С. Н.

Зав. кафедрой: к.ф.-м.н., доцент Черняева С. Н.