



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 «Технологии программирования»
(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 очная, заочная

г. Воронеж
2019

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

Рабочей программой дисциплины «Технологии программирования» 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
ОПК-1 Способен применять естественнонаучные и общепрофессиональные знания, методы математического анализа и моделирования, теоретического и экспериментального исследования профессиональной деятельности;	ИД-1ОПК-1	Знать: основы математики, физики, вычислительной техники и программирования.
	ИД-2ОПК-1	Уметь: решать стандартные профессиональные задачи с применением естественнонаучных и общепрофессиональных знаний, методов математического анализа и моделирования.
	ИД-3ОПК-1	Иметь навыки: теоретического и экспериментального исследования объектов профессиональной деятельности.
ОПК-3 Способен решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности;	ИД-1ОПК-3	Знать: принципы, методы и средства решения стандартных задач профессиональной деятельности на основе информационной и библиографической культуры с применением информационнокоммуникационных технологий и с учетом основных требований информационной безопасности.
	ИД-2ОПК-3	Уметь: решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационнокоммуникационных технологий и с учетом основных требований информационной безопасности.
	ИД-3ОПК-3	Иметь навыки: подготовки обзоров, аннотаций, составления рефератов, научных докладов, публикаций и библиографии по научноисследовательской работе с учетом требований информационной безопасности.
ОПК-6 Способен разрабатывать	ИД-1ОПК-6	Знать: методы алгоритмизации, языки и технологии программирования, пригодные

алгоритмы и программы, пригодные для практического применения в области информационных систем и технологий;		для практического применения в области информационных систем и технологий.
	ИД-2ОПК-6	Уметь: применять методы алгоритмизации, языки и технологии программирования при решении профессиональных задач в области информационных систем и технологий.
	ИД-3ОПК-6	Иметь навыки: программирования, отладки и тестирования прототипов программно-технических комплексов задач.

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	Введение в технологию программирования.	<i>ОПК-1, ОПК-3, ОПК-6</i>	<i>Тест текущего контроля 1 Контрольные вопросы по теме 1 экзамен</i>
2	Организация процесса проектирования программного обеспечения.	<i>ОПК-1, ОПК-3, ОПК-6</i>	<i>Тест текущего контроля 2 Контрольные вопросы по теме 2 экзамен</i>
3	Методы проектирования программного обеспечения.	<i>ОПК-1, ОПК-3, ОПК-6</i>	<i>Тест текущего контроля 3 Контрольные вопросы по теме 3 экзамен</i>
4	Парадигмы программирования.	<i>ОПК-1, ОПК-3, ОПК-6</i>	<i>Тест текущего контроля 4 Контрольные вопросы по теме 4 экзамен</i>
5	Технология создания программного кода.	<i>ОПК-1, ОПК-3, ОПК-6</i>	<i>Тест текущего контроля 5 экзамен</i>
6	Технологические средства разработки программного обеспечения.	<i>ОПК-1, ОПК-3, ОПК-6</i>	<i>Контрольные вопросы по теме 6 экзамен</i>
7	Методы отладки и тестирования программ.	<i>ОПК-1, ОПК-3, ОПК-6</i>	<i>Тест текущего контроля 7 экзамен</i>
8	Документирование программных средств.	<i>ОПК-1, ОПК-3, ОПК-6</i>	<i>Тест текущего контроля 8 Контрольные вопросы по теме 8 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			
ИД-1 ОПК-1 Знать: основы структурного, объектно-ориентированного и компонентного программирования.	<i>Отсутствие или фрагментарные представления об основах структурного, объектно-ориентированного и компонентного программирования.</i>	<i>Неполные представления об основах структурного, объектно-ориентированного и компонентного программирования.</i>	<i>Сформированные, но содержащие отдельные пробелы представления об основах структурного, объектно-ориентированного и компонентного программирования.</i>	<i>Сформированные систематические представления об основах структурного, объектно-ориентированного и компонентного программирования.</i>	<i>тестирование, ответы на контрольные вопросы, экзамен</i>
ИД-2 ОПК-1 Уметь: решать стандартные профессиональные задачи разработки программного обеспечения с применением общеинженерных знаний и методов системного анализа и моделирования.	<i>Отсутствие умений или фрагментарные умения решать стандартные задачи разработки программного обеспечения с применением общеинженерных знаний и методов системного анализа и моделирования.</i>	<i>В целом удовлетворительные, но не систематизированные умения решать стандартные профессиональные задачи разработки программного обеспечения с применением общеинженерных знаний и методов системного анализа и моделирования.</i>	<i>В целом удовлетворительные, но содержащие отдельные пробелы умения решать стандартные профессиональные задачи разработки программного обеспечения с применением общеинженерных знаний и методов системного анализа и моделирования.</i>	<i>Сформированные умения решать стандартные профессиональные задачи разработки программного обеспечения с применением общеинженерных знаний и методов системного анализа и моделирования.</i>	<i>Тестирование, ответы на контрольные вопросы, экзамен</i>
ИД-3 ОПК-1 Владеть: навыками теоретического и экспериментального исследования объектов профессиональной деятельности.	<i>Отсутствие владения или фрагментарные владения навыками теоретического и экспериментального исследования объектов профессиональной деятельности.</i>	<i>В целом удовлетворительные, но не систематизированные владения навыками теоретического и экспериментального исследования объектов профессиональной деятельности.</i>	<i>В целом удовлетворительные, но содержащие отдельные пробелы владения навыками теоретического и экспериментального исследования объектов профессиональной деятельности.</i>	<i>Сформированные владения навыками теоретического и экспериментального исследования объектов профессиональной деятельности.</i>	<i>Тестирование, ответы на контрольные вопросы, экзамен</i>
ИД-1 ОПК-3 Знать: принципы, методы и средства применения технологий программирования для решения	<i>Отсутствие или фрагментарные представления о принципах, методах и средствах применения технологий программирования</i>	<i>Неполные представления о принципах, методах и средствах применения технологий программирования для решения</i>	<i>Сформированные, но содержащие отдельные пробелы представления о принципах, методах и средствах применения</i>	<i>Сформированные систематические представления о принципах, методах и средствах применения</i>	<i>тестирование, ответы на контрольные вопросы, экзамен</i>

составления рефератов, научных докладов, публикаций и библиографии по научно-исследовательской работе с учетом требований информационной безопасности.	<i>аннотаций, составления рефератов, научных докладов, публикаций и библиографии по научно-исследовательской работе с учетом требований информационной безопасности.</i>	<i>заданий, отчетов, обзоров, аннотаций, составления рефератов, научных докладов, публикаций и библиографии по научно-исследовательской работе с учетом требований информационной безопасности.</i>	<i>технических заданий, отчетов, обзоров, аннотаций, составления рефератов, научных докладов, публикаций и библиографии по научно-исследовательской работе с учетом требований информационной безопасности.</i>	<i>аннотаций, составления рефератов, научных докладов, публикаций и библиографии по научно-исследовательской работе с учетом требований информационной безопасности.</i>	
ИД-1 ОПК-6 Знать: основные принципы и методы алгоритмизации, особенности языков и парадигм программирования, в также структурные компоненты и стадии технологии программирования, пригодные для практического применения в области информационных систем и технологий.	<i>Отсутствие или фрагментарные представления об основных принципах и методах алгоритмизации, особенностях языков и парадигм программирования, в также структурных компонентах и стадиях технологии программирования, пригодных для практического применения в области информационных систем и технологий.</i>	<i>Неполные представления об основных принципах и методах алгоритмизации, особенностях языков и парадигм программирования, в также структурных компонентах и стадиях технологии программирования, пригодных для практического применения в области информационных систем и технологий.</i>	<i>Сформированные, но содержащие отдельные пробелы представления об основных принципах и методах алгоритмизации, особенностях языков и парадигм программирования, в также структурных компонентах и стадиях технологии программирования, пригодных для практического применения в области информационных систем и технологий.</i>	<i>Сформированные систематические представления об основных принципах и методах алгоритмизации, особенностях языков и парадигм программирования, в также структурных компонентах и стадиях технологии программирования, пригодных для практического применения в области информационных систем и технологий.</i>	<i>тестирование, ответы на контрольные вопросы, экзамен</i>
ИД-2 ОПК-6 Уметь: применять методы алгоритмизации, языки и технологии программирования при решении профессиональных задач в области информационных систем и технологий.	<i>Отсутствие умений применять методы алгоритмизации, языки и технологии программирования при решении профессиональных задач в области информационных систем и технологий.</i>	<i>В целом удовлетворительные, но не систематизированные умения применять методы алгоритмизации, языки и технологии программирования при решении профессиональных задач в области информационных систем и технологий.</i>	<i>В целом удовлетворительные, но содержащие отдельные пробелы умения применять методы алгоритмизации, языки и технологии программирования при решении профессиональных задач в области информационных систем и технологий.</i>	<i>Сформированные умения применять методы алгоритмизации, языки и технологии программирования при решении профессиональных задач в области информационных систем и технологий.</i>	<i>Тестирование, ответы на контрольные вопросы, экзамен</i>
ИД-3 ОПК-6	<i>Отсутствие</i>	<i>В целом</i>	<i>В целом</i>	<i>Сформированные</i>	<i>Тестирование,</i>

Владеть: навыками выбора и использования технологии программирования, модели жизненного цикла программного обеспечения, языка программирования, среды отладки и тестирования прототипов программно-технических комплексов задач.	<i>владения или фрагментарные владения навыками выбора и использования технологии программирования, модели жизненного цикла программного обеспечения, языка программирования, среды отладки и тестирования прототипов программно-технических комплексов задач.</i>	<i>удовлетворительные, но не систематизированные владения навыками выбора и использования технологии программирования, модели жизненного цикла программного обеспечения, языка программирования, среды отладки и тестирования прототипов программно-технических комплексов задач.</i>	<i>удовлетворительные, но содержащие отдельные пробелы владения навыками выбора и использования технологии программирования, модели жизненного цикла программного обеспечения, языка программирования, среды отладки и тестирования прототипов программно-технических комплексов задач.</i>	<i>е владения навыками выбора и использования технологии программирования, модели жизненного цикла программного обеспечения, языка программирования, среды отладки и тестирования прототипов программно-технических комплексов задач.</i>	<i>ответы на контрольные вопросы, экзамен</i>
--	--	---	---	---	---

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ ТЕКУЩЕГО КОНТРОЛЯ

Тесты и контрольные вопросы текущего контроля

Тема 1

Введение в технологию программирования.

Тест текущего контроля 1

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

1. Какие программы можно отнести к системному программному обеспечению:
 - a) операционные системы;
 - b) прикладные программы;
 - c) игровые программы.

2. Какие программы можно отнести к системному ПО:
 - a) драйверы;
 - b) текстовые редакторы;
 - c) электронные таблицы;
 - d) графические редакторы.

3. Специфические особенности ПО как продукта:
 - a) продажа по ценам ниже себестоимости (лицензирование);
 - b) низкие материальные затраты при создании программ;
 - c) возможность создание программ небольшие коллективом или даже одним человеком;
 - d) разнообразие решаемых задач с помощью программных средств.

4. Какие программы можно отнести к системному ПО:
- a) программа расчета заработной платы;
 - b) электронные таблицы;
 - c) СУБД (системы управления базами данных).
5. Какие программы нельзя отнести к системному ПО:
- a) игровые программы;
 - b) компиляторы языков программирования;
 - c) операционные системы;
 - d) системы управления базами данных.
6. Какие программы можно отнести к прикладному программному обеспечению:
- a) электронные таблицы;
 - b) таблицы решений;
 - c) СУБД (системы управления базами данных).
7. Какие программы можно отнести к прикладному ПО:
- a) программа расчета заработной платы;
 - b) диспетчер программ;
 - c) программа «Проводник» (Explorer).
8. Какие программы нельзя отнести к прикладному ПО:
- a) компиляторы и (или) интерпретаторы;
 - b) текстовые и (или) графические редакторы;
 - c) электронные таблицы.
9. Можно ли отнести операционную систему к программному обеспечению:
- a) да;
 - b) нет.
10. Можно ли отнести операционную систему к прикладному программному обеспечению:
- a) да;
 - b) нет.
11. Специфические особенности ПО как продукта:
- a) низкие затраты при дублировании;
 - b) универсальность;
 - c) простота эксплуатации;
 - d) наличие поддержки (сопровождения) со стороны разработчика.
12. Какие программы можно отнести к системному ПО:
- a) утилиты;
 - b) экономические программы;

- c) статистические программы;
- d) мультимедийные программы.

13. Этап, занимающий наибольшее время, в жизненном цикле программы:

- a) сопровождение;
- b) проектирование;
- c) тестирование;
- d) программирование;
- e) формулировка требований.

14. Этап, занимающий наибольшее время, при разработке программы:

- a) тестирование;
- b) сопровождение;
- c) проектирование;
- d) программирование;
- e) формулировка требований.

15. Первый этап в жизненном цикле программы:

- a) формулирование требований;
- b) анализ требований;
- c) проектирование;
- d) автономное тестирование;
- e) комплексное тестирование.

16. Один из необязательных этапов жизненного цикла программы:

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

17. Самый большой этап в жизненном цикле программы:

- a) эксплуатация;
- b) изучение предметной области;
- c) программирование;
- d) тестирование;
- e) корректировка ошибок.

18. Какой этап выполняется раньше:

- a) отладка;
- b) тестирование.

19. Какой этап выполняется раньше:

- a) отладка;
- b) оптимизация;
- c) программирование;

d) тестирование.

20. Что выполняется раньше:

- a) компиляция;
- b) отладка;
- c) компоновка;
- d) тестирование.

21. Что выполняется раньше:

- a) проектирование;
- b) программирование;
- c) отладка;
- d) тестирование.

Контрольные вопросы по теме 1:

1. Какими условиями определяется успешность бизнеса?
2. Дайте определение понятия технология.
3. Дайте определение понятия технология программирования.
4. Дайте определение понятия программная инженерия.
5. Что входит в ядро знаний SWEBOOK?
6. Какие специалисты заняты в программной инженерии?
7. Охарактеризуйте ТП 50-х годов XX века.
8. Охарактеризуйте ТП 60-х годов XX века.
9. Охарактеризуйте ТП 70-х годов XX века.
10. Охарактеризуйте ТП 80-х годов XX века.
11. Охарактеризуйте ТП 90-х годов XX века.
12. Охарактеризуйте ТП XXI века.
13. Что такое жизненный цикл ПС?

Тема 2

Организация процесса проектирования программного обеспечения

Тест текущего контроля 2

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.

Контрольные вопросы по теме 2:

1. Основное назначение моделей жизненного цикла (ЖЦ) ПС?
2. Перечислите основные процессы ЖЦ ПС.
3. Назовите вспомогательные процессы ЖЦ ПС.
4. Опишите структуру стандарта ГОСТ ISO/IEC 12207.
5. Перечислите основные подходы организации процессов создания ПС и назовите основные виды моделей ЖЦ ПС.
6. Опишите суть водопадного подхода разработки ПС.
7. Опишите суть исследовательского программирования.
8. Опишите суть прототипирования при разработке ПС.
9. Опишите основные черты подходов формальных преобразований и сборочного программирования при разработке ПС.
10. Какие общие черты имеют инкрементная и эволюционная модели?
11. Как построить новую модель ЖЦ на основе стандарта ISO/IEC 12207?

Тема 3

Методы проектирования программного обеспечения

Тест текущего контроля 3

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.

Контрольные вопросы по теме 3:

1. Назначение *внешнего описания ПС?*
2. С чего начинается *разработка внешнего описания ПС?*
3. Перечислите способы *разработки определения требований к ПС.*
4. Что входит в *цикл работы с требованиями?*
5. Задачи *системного анализа в процессе определения требований?*
6. Из чего состоят *требования к ПС?* Классификация требований.
7. Перечислите *свойства требований к ПС.*
8. Какие существуют варианты *формализации требований к ПС?*
9. Какова *структура внешнего описания?*
10. Перечислите основные *стадии разработки ПС и этапы работ* согласно ГОСТ 19.102 ЕСПД.
11. Перечислите основные *разделы технического задания на разработку ПС* согласно ГОСТ 19.201 ЕСПД.
12. Какие подразделы должен содержать раздел ТЗ *«Требования к программе»* согласно ГОСТ 19.201 ЕСПД?
13. Что означают примитивы качества *завершенность, точность?*
14. Что означают примитивы качества *автономность, независимость от устройств?*
15. Что означают примитивы качества *устойчивость, защищенность?*

16. Что означают примитивы качества *П-документированность*, *С-документированность*?
17. Что означают примитивы качества *информативность*, *коммуникабельность*?
18. Какие примитивы качества характеризуют *эффективность ПС*?
19. От каких примитивов качества зависит *надежность ПС*?
20. От каких примитивов качества зависит *легкость применения ПС*?
21. От каких примитивов качества зависит *мобильность ПС*?
22. Структура *функциональной спецификации внешнего описания ПС*?

Тема 4

Парадигмы программирования

Тест текущего контроля 4

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 модульном 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.

Контрольные вопросы по теме 4:

1. В чем заключается *цель модульного программирования*?
2. Назовите основные *характеристики программного модуля*.
3. Что определяет *связность модуля*?
4. Перечислите названия модулей с разными *степенями связности* по степени их возрастания.
5. Что такое *сцепление модуля*?
6. Какой модуль называется *рутинным*? Какой модуль зависит от *предыстории*?
7. Какая *модульная структура программы* используется в ТП?
8. Перечислите *классические методы разработки* структуры программы. Назовите предпочтительный метод.
9. Суть *конструктивного подхода* разработки структуры программы?
10. Суть *целенаправленной конструктивной реализации* разработки структуры программы?
11. Суть *архитектурного подхода* разработки структуры программы?
12. Какие существуют *методы контроля* структуры программы?
13. Перечислите *шаги разработки программного модуля*.
14. Основная суть *структурного программирования*?
15. Основные *конструкции структурного программирования*?
16. Почему в структурном программировании не рекомендуется использовать *оператор GOTO*?
17. Когда *оператор GOTO* рекомендуется использовать?
18. Почему построение *блок-схем* не рекомендуется при программировании модуля?
19. В каких случаях построение *блок-схем* эффективно при программировании модуля?
20. Суть *метода пошаговой детализации* при построении текста модуля?

21. Что описывается на *первом шаге пошаговой детализации* при построении текста модуля?
22. Чем завершается *метод пошаговой детализации* построения текста модуля?
23. В чем заключается *статическая проверка текста модуля*?
24. Суть *сквозного прослеживания текста модуля*?

Тема 5

Технология создания программного кода

Тест текущего контроля 5

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:
- САПР (систем автоматизированного проектирования);
 - СУБД (систем управления базами данных);
 - ОС (операционных систем).

Тема 6

Технологические средства разработки программного обеспечения

Тест текущего контроля 6

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

1. Что такое автоматизация программирования:
 - а) создание исходного кода программными средствами;
 - б) создание исходного кода при помощи компилятора;
 - в) создание исходного кода без разработки алгоритма.
2. В чем сущность автоматизации программирования:
 - а) создание программы без написания ее текста;
 - б) получение готовой программы без выполнения компоновки;
 - в) в отсутствии компиляции.
3. Возможна ли автоматизация программирования:
 - а) да;
 - б) нет.
4. Создание исполняемого кода программы без написания исходного кода называется:
 - а) составлением спецификаций;
 - б) отладкой;
 - в) проектированием.
 - д) автоматизацией программирования;
5. Одно из преимуществ автоматизации программирования:
 - а) наглядное программирование с визуальным контролем;
 - б) получение стандартной программы;
 - в) создание программы с оптимальным кодом.
6. Один из методов автоматизации программирования:
 - а) структурное программирование;
 - б) модульное программирование;
 - в) визуальное программирование;
 - д) объектно-ориентированное программирование.
7. Влияет ли автоматизация программирования на эффективность программы:
 - а) 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.

Методы отладки и тестирования программ

Тест текущего контроля 7

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;

е) оформление авторских прав на программу.

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

- а) синтаксическая;
- б) семантическая;
- с) логическая;
- д) символная.

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

- а) семантическая;
- б) синтаксическая;
- с) логическая;
- д) символная.

11. Ошибки при написании программы бывают:

- а) синтаксические;
- б) орфографические;
- с) лексические;
- д) фонетические;
- е) морфологические.

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

- а) отладка;
- б) тестирование;
- с) компоновка;
- д) транзакция;
- е) трансляция.

13. Программа для просмотра значений переменных при выполнении программы:

- а) отладчик;
- б) компилятор;
- с) интерпретатор;
- д) трассировка;
- е) тестирование.

14. Отладка – это:

- а) процедура поиска ошибок, когда известно, что ошибка есть;
- б) определение списка параметров;
- с) правило вызова процедур (функций);
- д) составление блок-схемы алгоритма.

15. Когда программист может проследить последовательность выполнения команд программы:

- а) при трассировке;
- б) при тестировании;
- с) 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.

Контрольные вопросы по теме 7:

1. Что такое отладка?
2. Основные задачи тестирования?
3. Основные стратегии проектирования тестов?
4. Что такое оптимальная стратегия проектирования тестов?
5. Какие существуют виды тестирования?
6. Что такое автономная отладка?
7. Какие существуют разновидности автономной отладки?
8. Что такое восходящее тестирование?
9. Что такое нисходящее тестирование?
10. Как осуществляется тестирование сопряжения модулей?
11. Какие существуют этапы комплексной отладки ПС?
12. Перечислите заповеди отладки ПС.

Тема 8

Документирование программных средств

Контрольные вопросы по теме 8:

1. Что такое *процесс сопровождения ПС*?
2. Какие *нормативные документы регламентируют процесс сопровождения*?
3. В чем *сопровождение ПС похоже на разработку ПС*?
4. В чем заключается *уникальность работы сопровождения*?
5. Перечислите *дополнительные работы процесса сопровождения*.
6. Какие существуют *уровни планирования сопровождения*?
7. Как осуществляется *планирование версий*?
8. Что такое *конфигурационное управление*?
9. Что такое *эволюция ПС*?
10. Какие существуют *технологии эволюции ПС*?
11. Что такое *реинжиниринг и рефакторинг*?
12. Что такое *реверсная инженерия*?

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».

Критерии оценки ответов на контрольные вопросы

Критерии оценивания	Результат
<i>Ответ на вопрос дан полный, правильный, аргументированный.</i>	5
<i>Ответ на вопрос дан неполный или студент допустил неточность.</i>	4
<i>Ответ дан частично верный.</i>	3
<i>Ответ на контрольный вопрос дан неверный.</i>	2

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОГО КОНТРОЛЯ

Промежуточная аттестация – экзамен (в тестовой форме).

ИТОГОВЫЙ ТЕСТ ПО ДИСЦИПЛИНЕ

1. Какие программы можно отнести к системному программному обеспечению:
 - a) операционные системы;
 - b) прикладные программы;
 - c) игровые программы.

2. Какие программы можно отнести к системному ПО:
 - a) драйверы;
 - b) текстовые редакторы;
 - c) электронные таблицы;
 - d) графические редакторы.

3. Специфические особенности ПО как продукта:
 - a) продажа по ценам ниже себестоимости (лицензирование);
 - b) низкие материальные затраты при создании программ;
 - c) возможность создание программ небольшие коллективом или даже одним человеком;
 - d) разнообразие решаемых задач с помощью программных средств.

4. Какие программы можно отнести к системному ПО:
 - a) программа расчета заработной платы;
 - b) электронные таблицы;

- c) СУБД (системы управления базами данных).
5. Какие программы нельзя отнести к системному ПО:
- a) игровые программы;
 - b) компиляторы языков программирования;
 - c) операционные системы;
 - d) системы управления базами данных.
6. Какие программы можно отнести к прикладному программному обеспечению:
- a) электронные таблицы;
 - b) таблицы решений;
 - c) СУБД (системы управления базами данных).
7. Какие программы можно отнести к прикладному ПО:
- a) программа расчета заработной платы;
 - b) диспетчер программ;
 - c) программа «Проводник» (Explorer).
8. Какие программы нельзя отнести к прикладному ПО:
- a) компиляторы и (или) интерпретаторы;
 - b) текстовые и (или) графические редакторы;
 - c) электронные таблицы.
9. Можно ли отнести операционную систему к программному обеспечению:
- a) да;
 - b) нет.
10. Можно ли отнести операционную систему к прикладному программному обеспечению:
- a) да;
 - b) нет.
11. Специфические особенности ПО как продукта:
- a) низкие затраты при дублировании;
 - b) универсальность;
 - c) простота эксплуатации;
 - d) наличие поддержки (сопровождения) со стороны разработчика.
12. Какие программы можно отнести к системному ПО:
- a) утилиты;
 - b) экономические программы;
 - c) статистические программы;
 - d) мультимедийные программы.
13. Этап, занимающий наибольшее время, в жизненном цикле программы:

- a) сопровождение;
- b) проектирование;
- c) тестирование;
- d) программирование;
- e) формулировка требований.

14. Этап, занимающий наибольшее время, при разработке программы:

- a) тестирование;
- b) сопровождение;
- c) проектирование;
- d) программирование;
- e) формулировка требований.

15. Первый этап в жизненном цикле программы:

- a) формулирование требований;
- b) анализ требований;
- c) проектирование;
- d) автономное тестирование;
- e) комплексное тестирование.

16. Один из необязательных этапов жизненного цикла программы:

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

17. Самый большой этап в жизненном цикле программы:

- a) эксплуатация;
- b) изучение предметной области;
- c) программирование;
- d) тестирование;
- e) корректировка ошибок.

18. Какой этап выполняется раньше:

- a) отладка;
- b) тестирование.

19. Какой этап выполняется раньше:

- a) отладка;
- b) оптимизация;
- c) программирование;
- d) тестирование.

20. Что выполняется раньше:

- a) компиляция;

- b) отладка;
- c) компоновка;
- d) тестирование.

21. Что выполняется раньше:

- a) проектирование;
- b) программирование;
- c) отладка;
- d) тестирование.

22. В стадии разработки программы не входит:

- a) автоматизация программирования;
- b) постановка задачи;
- c) составление спецификаций;
- d) эскизный проект;
- e) тестирование.

23. Самый важный критерий качества программы:

- a) работоспособность;
- b) надежность;
- c) эффективность;
- d) быстродействие;
- e) простота эксплуатации.

24. Способы оценки качества:

- a) сравнение с аналогами;
- b) наличие документации;
- c) оптимизация программы;
- d) структурирование алгоритма.

25. Существует ли связь между эффективностью и оптимизацией программы:

- a) да;
- b) нет.

26. Наиболее важный критерий качества:

- a) надежность;
- b) быстродействие;
- c) удобство в эксплуатации;
- d) удобный интерфейс;
- e) эффективность.

27. Способы оценки надежности:

- a) тестирование;
- b) сравнение с аналогами;
- c) трассировка;
- d) оптимизация.

28. Повышает ли качество программ оптимизация:
- a) да;
 - b) нет.
29. Существует ли связь между надежностью и быстродействием:
- a) нет;
 - b) да.
30. В каких единицах можно измерить надежность:
- a) отказов/час;
 - b) час;
 - c) Кбайт/сек;
 - d) операций/сек.
31. В каких единицах можно измерить быстродействие:
- a) отказов/час;
 - b) км/час;
 - c) Кбайт/сек;
 - d) операций/сек.
32. Что относится к этапу программирования:
- a) написание кода программы;
 - b) разработка интерфейса;
 - c) работоспособность;
 - d) анализ требований.
33. Последовательность этапов программирования:
- a) компилирование, компоновка, отладка;
 - b) компоновка, отладка, компилирование;
 - c) отладка, компилирование, компоновка;
 - d) компилирование, отладка, компоновка.
- 34) Инструментальные средства программирования:
- a) компиляторы, интерпретаторы;
 - b) СУБД (системы управления базами данных);
 - c) BIOS (базовая система ввода-вывода);
 - d) ОС (операционные системы).
35. На языке программирования составляется:
- a) исходный код;
 - b) исполняемый код;
 - c) объектный код;
 - d) алгоритм.
36. Правила, которым должна следовать программа это:

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

37. Можно ли внутри цикла поместить еще один цикл:

- a) да;
- b) нет.

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

- a) да;
- b) нет.

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

- a) да;
- b) нет.

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

- a) нет;
- b) да.

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

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

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

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

43. Методы программирования (укажите НЕ верный ответ):

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

44. Что выполняется раньше:

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

45. Можно ли переменным присваивать произвольные идентификаторы:
- a) да;
 - b) нет.
46. 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.
47. Какие символы не допускаются в именах переменных:
- a) пробелы;
 - b) цифры;
 - c) подчеркивание
48. Можно ли использовать имена, которые уже были использованы в другой программе (модуле):
- a) да;
 - b) нет.
49. Можно ли ставить знак подчеркивания в начале имени:
- a) да, но не рекомендуется;
 - b) да, без ограничений;
 - c) нет.
50. Как называется способ составления имен переменных, когда в начале имени сообщается тип переменной:
- a) прямым указанием;
 - b) венгерской нотацией;
 - c) структурным программированием;
 - d) поляризацией.
51. Можно ли писать комментарии в отдельной строке:
- a) да;
 - b) нет.
52. Транслируются ли комментарии:
- a) да;
 - b) нет.
53. Наличие комментариев позволяет:
- a) быстрее найти ошибки в программе;
 - b) быстрее писать программы;
 - c) быстрее выполнять программы.

54. Наличие комментариев позволяет:
- a) легче разобраться в программе;
 - b) применять сложные структуры;
 - c) увеличить быстродействие.
55. Наличие комментариев позволяет:
- a) улучшить читабельность программы;
 - b) улучшить эксплуатацию программы;
 - c) повысить надежность программы.
56. Что определяет выбор языка программирования:
- a) область приложения;
 - b) знание языка;
 - c) наличие дополнительных библиотек.
57. Возможно ли комбинирование языков программирования в рамках одной задачи:
- a) да;
 - b) нет.
58. Допустимо ли комбинирование языков программирования в рамках одной задачи :
- a) да;
 - b) нет.
59. 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.
60. Для каких задач характерен большой объем вычислений, использование сложного математического аппарата:
- a) для инженерных задач;
 - b) для системных задач;
 - c) для экономических задач.
61. На каком этапе производится выбор языка программирования:
- a) проектирование;
 - b) программирование;
 - c) отладка;
 - d) тестирование.

62. Можно ли использовать комбинацию языков программирования в рамках одного проекта:
- да;
 - нет.
63. Для решения экономических задач характерно применение:
- СУБД (систем управления базами данных);
 - языков высокого уровня;
 - языков низкого уровня;
 - применение сложных математических расчетов.
64. Для решения инженерных задач характерно применение:
- САПР (систем автоматизированного проектирования);
 - СУБД (систем управления базами данных);
 - ОС (операционных систем).
65. Причины синтаксических ошибок:
- плохое знание языка программирования;
 - ошибки в исходных данных;
 - ошибки, допущенные на более ранних этапах;
 - неправильное применение процедуры тестирования.
66. Когда можно обнаружить синтаксические ошибки:
- при компиляции;
 - при отладке;
 - при тестировании;
 - на этапе проектирования;
 - при эксплуатации.
67. Ошибки компоновки заключаются в том, что:
- указано внешнее имя, но не объявлено;
 - неправильно использовано зарезервированное слово;
 - составлено неверное выражение;
 - указан неверный тип переменной.
68. Могут ли проявиться ошибки при изменении условий эксплуатации:
- да;
 - нет.
69. Могут ли проявиться ошибки при изменении в предметной области:
- да;
 - нет.
70. Возможно ли программирование с защитой от ошибок:
- да;

b) нет.

71. Есть ли недостатки программирования с защитой от ошибок:

- a) да;
- b) нет.

72. 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.

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

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

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

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

75. Ошибки при написании программы бывают:

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

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

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

77. Программа для просмотра значений переменных при выполнении программы:

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

78. 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.

79. Когда программист может проследить последовательность выполнения команд программы:

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

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

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

81. 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,

82. Тестирование бывает:

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

83. Тестирование бывает:

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

84. Существует ли различие между отладкой и тестированием:

- a) да;
- b) нет.

85. При комплексном тестировании проверяются:

- a) согласованность работы отдельных частей программы;

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

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

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

87. Процесс исполнения программы с целью обнаружения ошибок:

- a) тестирование;
- b) кодирование;
- c) сопровождение;
- d) проектирование.

88. Автономное тестирование это:

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

89. Трассировка это:

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

90. Локализация ошибки:

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

91. Назначение тестирования:

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

92. Назначение отладки:

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

93. Инструментальные средства отладки (НЕ правильный ответ):
- a) компиляторы;
 - b) отладчики;
 - c) трассировка.
94. Отладка программ это:
- a) локализация и исправление ошибок;
 - b) алгоритмизация программирования;
 - c) компиляция и компоновка.
95. Что выполняется раньше, автономная или комплексная отладка:
- a) автономная;
 - b) комплексная.
96. Что выполняется раньше, отладка или тестирование:
- a) отладка;
 - b) тестирование.
97. Что такое автоматизация программирования:
- a) создание исходного кода программными средствами;
 - b) создание исходного кода при помощи компилятора;
 - c) создание исходного кода без разработки алгоритма.
98. В чем сущность автоматизации программирования:
- a) создание программы без написания ее текста;
 - b) получение готовой программы без выполнения компоновки;
 - c) в отсутствии компиляции.
99. Возможна ли автоматизация программирования:
- a) да;
 - b) нет.
100. Создание исполняемого кода программы без написания исходного кода называется:
- a) составлением спецификаций;
 - b) отладкой;
 - c) проектированием.
 - d) автоматизацией программирования.
101. Одно из преимуществ автоматизации программирования:
- a) наглядное программирование с визуальным контролем;
 - b) получение стандартной программы;
 - c) создание программы с оптимальным кодом.

102. Один из методов автоматизации программирования:
- a) структурное программирование;
 - b) модульное программирование;
 - c) визуальное программирование;
 - d) объектно-ориентированное программирование.
103. Влияет ли автоматизация программирования на эффективность программы:
- a) нет;
 - b) да.
104. Автоматизация программирования позволяет:
- a) повысить надежность программы;
 - b) сократить время разработки программы;
 - c) повысить быстродействие программы.
105. Позволяет ли автоматизация программирования всегда создавать эффективные программы:
- a) да.
 - b) нет.
106. Позволяет ли автоматизация программирования всегда создавать надежные программы:
- a) нет;
 - b) да.
107. Недостаток автоматизации программирования;
- a) низкое быстродействие;
 - b) большой размер программы;
 - c) сложность программы.
108. Возможны ли ошибки при автоматизации программирования:
- a) да;
 - b) нет.
109. Нужно ли выполнять тестирование при автоматизации программирования:
- a) да;
 - b) нет.
110. Выполняется ли процедура компиляции при автоматизации программирования:
- a) да;
 - b) нет.

111. Что легко поддается автоматизации:
- a) интерфейс;
 - b) работа с файлами;
 - c) сложные логические задачи;
 - d) алгоритмизация.
112. Относится ли визуальное программирование к средствам автоматизации:
- a) да;
 - b) нет.
113. Нахождение наилучшего варианта из множества возможных:
- a) оптимизация;
 - b) тестирование;
 - c) автоматизация;
 - d) отладка;
 - e) сопровождение.
114. 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.
115. Критерии оптимизации:
- a) время выполнения или размер требуемой памяти;
 - b) размер программы и ее эффективность;
 - c) независимость модулей;
 - d) качество программы, ее надежность.
116. Критерии оптимизации:
- a) эффективность использования ресурсов;
 - b) структурирование алгоритма;
 - c) структурирование программы.
117. Возможна ли оптимизация программ без участия программиста:
- a) да;
 - b) нет.
118. Возможна ли оптимизация циклов:
- a) да;
 - b) нет.
119. 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.

120. Оптимизация циклов заключается в:

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

121. Оптимизация программы это:

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

122. Критерии оптимизации программы:

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

123. Результат оптимизации программы:

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

124. 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.

125. 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 функцию.

126. Можно ли сочетать модульное и структурное программирование:

- a) да;
- b) нет.

127. Может ли модуль включать несколько процедур или функций:

- a) да;

b) нет.

128. Рекомендуемые размеры модулей:

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

129. 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.

130. При модульном программировании желательно, чтобы модуль имел:

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

131. Модульное программирование это:

- a) разбиение программы на отдельные части;
- b) структурирование;
- c) использование стандартных процедур и функций.

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

- a) можно;
- b) нельзя.

133. Разрешается ли использование циклов при модульном программировании:

- a) да;
- b) нет.

134. Разрешается ли использование условных операторов при модульном программировании:

- a) да;
- b) нет.

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

- a) нет;
- b) да.

136. 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.

138. 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.

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

- a) нет;
- b) да.

140. Можно ли сочетать структурное программирование с модульным:

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

141. Любую ли программу можно привести к структурированному виду:

- a) любую;
- b) не все;
- c) нельзя.

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

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

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

- a) да;
- b) нет.

144. Возможно ли программирование без оператора GO TO:

- a) да;
- b) нет.

145. При структурном программировании задача выполняется:

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

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

- a) нет;
- b) да;
- c) иногда.

147. Разрешается ли использование циклов при структурном программировании:

- a) да;
- b) нет.

148. Разрешается ли использование оператора IF при структурном программировании:

- a) да;
- b) нет.

149. Программирование без GO TO применяется. при:

- a) структурном программировании;
- b) модульном программировании;
- c) объектно-ориентированном программировании;
- d) все ответы верные.

150. 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.

Критерии оценки результатов итогового 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 – курсовая работа.

Тема 1. Разработка приложения для предметной области «Учёт клиентов и мобильных телефонов компании, предоставляющей услуги мобильной связи».

Тема 2. Разработка приложения для предметной области «Учёт товаров в магазине».

Тема 3. Разработка приложения для предметной области «Организация учебного процесса в ВУЗе».

Тема 4. Разработка приложения для предметной области «Аквариум».

Тема 5. Разработка приложения для предметной области «Учёт средств вычислительной техники на предприятии».

Тема 6. Разработка приложения для предметной области «Реестр типов речных и морских транспортных средств».

Тема 7. Разработка приложения для предметной области «Учёт транспортных средств и их владельцев».

Тема 8. Разработка приложения для предметной области «Учёт рек с точки зрения народнохозяйственного значения».

Тема 9. Разработка приложения для предметной области «Учёт растений в ботаническом саду».

Тема 10. Разработка приложения для предметной области «Учёт диких животных в охотоведческом хозяйстве».

Тема 11. Разработка приложения для предметной области «Учёт домашних животных в фермерском хозяйстве».

Тема 12. Разработка приложения для предметной области «Сведения о памятниках истории и архитектуры».

Тема 13. Разработка приложения для предметной области «Сбор сведений о писателях и их литературных произведениях».

Тема 14. Разработка приложения для предметной области «Сведения о промышленных предприятиях».

Тема 15. Разработка приложения для предметной области «Учёт абитуриентов, поступающих в ВУЗ».

Тема 16. Разработка приложения для предметной области «Математика».

Тема 17. Разработка приложения для предметной области «Учёт вкладов, помещённых в банк».

Тема 18. Разработка приложения для предметной области «Учёт пациентов в регистратуре поликлиники».

Тема 19. Разработка приложения для предметной области «Учёт объектов библиотечного фонда».

Тема 20. Разработка приложения для предметной области «Учёт заказов клиентов в ресторане».

Критерии оценки курсовой работы

Таблица 5

Критерии оценки

Наименование показателя	Критерии оценки	Максимальное количество баллов	Количество баллов
I. КАЧЕСТВО РАБОТЫ			
<p>1 Соответствие содержания работы заданию, степень раскрытия темы. Обоснованность и доказательность выводов</p>	<p>-соответствие содержания теме курсовой работы; - умение работать с литературой, систематизировать и структурировать материал; - умение обобщать, сопоставлять различные точки зрения по рассматриваемому вопросу, аргументировать основные положения и выводы -уровень владения тематикой и научное значение исследуемого вопроса</p>	10	
<p>2. Грамотность изложения и качество оформления работы</p>	<p>- правильное оформление ссылок на используемую литературу; - грамотность и культура изложения; - владение терминологией и понятийным аппаратом проблемы; - соблюдение требований к объему курсовой работы; - отсутствие орфографических и синтаксических ошибок, стилистических погрешностей.</p>	5	
<p>3. Самостоятельность выполнения работы, глубина проработки материала, использование рекомендованной и справочной литературы</p>	<p>- степень знакомства автора работы с актуальным состоянием изучаемой проблематики; - полнота цитирования источников, степень использования в работе результатов исследований и установленных научных фактов. - дополнительные знания, использованные при написании работы, которые получены помимо предложенной образовательной программы; - новизна поданного материала и</p>	5	

	рассмотренной проблемы.		
Общая оценка за выполнение		20	
ОТВЕТЫ НА ДОПОЛНИТЕЛЬНЫЕ ВОПРОСЫ ПО СОДЕРЖАНИЮ РАБОТЫ			
Вопрос 1		5	
Вопрос 2		5	
Общая оценка за ответы на вопросы		10	
Итого		30	

Для перевода баллов критериально-шкалированной таблицы в оценку применяется универсальная шкала оценки образовательных достижений. Если студент набирает 27-30 баллов и выше - оценка «отлично», 26 -21 баллов и выше - оценка «хорошо», 18-21 баллов и выше - оценка «удовлетворительно», менее 18 - оценка «не зачтено».

Составитель: ст. преподаватель Плотников С. Н.

Зав. кафедрой: д.т.н., профессор Лапшина М. Л.

Рабочая программа рассмотрена на заседании кафедры математики, информационных систем и технологий и утверждена на 2019/2020 учебный год.
 Протокол № 9 от 22 мая 2019.