



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*

**МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ
ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ**

по дисциплине

ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ

**Для студентов, обучающихся по направлению
09.03.02 – “Информационные системы и технологии”,
очной, очно-заочной, заочной форм обучения**

г. Воронеж
2023

Методические рекомендации для самостоятельной работы по дисциплине «Технологии программирования» / Сост. С. Н. Черняева. - Воронеж: Воронежский филиал ФГБОУ ВО «ГУМРФ имени адмирала С.О. Макарова», 2023. - 28 с. – Текст : непосредственный.

Методические рекомендации для самостоятельной работы составлены в соответствии с программой дисциплины «Технологии программирования», изучаемой в Воронежском филиале ФГБОУ ВО «ГУМРФ имени адмирала С.О. Макарова. Рекомендации предназначены для организации контактной работы с обучающимися по дисциплине «Технологии программирования», а также для самостоятельной внеаудиторной работы обучающихся.

Методические рекомендации утверждены на заседании кафедры математики, информационных систем и технологий Воронежского филиала ФГБОУ ВО «ГУМРФ имени адмирала С.О. Макарова» 29.06.2023 г., протокол № 10.

© ВФ ФГБОУ ВО «ГУМРФ имени адмирала С. О. Макарова», 2023

© Черняева С. Н., 2023

Содержание

1. Цели и задачи дисциплины.....	6
2. Методические указания по изучению дисциплины «Технологии программирования»	6
2.1. Методические рекомендации по подготовке к лекциям	6
2.2. Методические рекомендации по подготовке к лабораторным занятиям.....	9
3. Методические рекомендации по организации самостоятельной работы обучающихся по дисциплине «Технологии программирования».....	12
3.1. Общие методические рекомендации по самостоятельной работе	12
3.2. Методические рекомендации по курсовой работе.....	13
4. Промежуточная аттестация	18
5. Перечень основной, дополнительной учебной литературы и учебно-методической литературы для самостоятельной работы обучающихся, необходимой для освоения дисциплины	26

Введение

Для успешного освоения учебной дисциплины обучающимся необходимо изучить лекционный материал и рекомендуемую литературу, отработать изученный материал на практических занятиях, выполнить задания для самостоятельной работы. Практические занятия проводятся с целью закрепления лекционного материала, овладения понятийным аппаратом предмета, методами работы, изучаемыми в рамках учебной дисциплины.

Все формы практических занятий (семинары – практикумы, практические, лабораторные) направлены на практическое усвоение теоретических знаний, полученных на лекциях. Главной целью такого рода занятий является: научить студентов применению теоретических знаний на практике. С этой целью на занятиях моделируются фрагменты их будущей деятельности в виде учебных ситуационных задач, при решении которых студенты отрабатывают различные действия по применению соответствующих практических навыков.

Самостоятельная работа студента – это планируемая учебная, учебно-исследовательская, научно-исследовательская работа студентов, выполняемая во внеаудиторное (аудиторное) время по заданию и при методическом руководстве преподавателя, но без его непосредственного участия (при частичном непосредственном участии преподавателя, оставляющем ведущую роль за работой студентов).

Самостоятельная работа в современном образовательном процессе рассматривается как форма организации обучения, которая способна обеспечивать самостоятельный поиск необходимой информации, творческое восприятие и осмысление учебного материала в ходе аудиторных занятий, разнообразные формы познавательной деятельности студентов на занятиях и во внеаудиторное время, развитие аналитических способностей, навыков контроля и планирования учебного времени, выработку умений и навыков рациональной организации учебного труда. Таким образом, самостоятельная работа – форма организации образовательного процесса, стимулирующая активность,

самостоятельность, познавательный интерес студентов.

Самостоятельная работа обучающихся является важным видом учебной и научной деятельности студента. Самостоятельная работа студентов играет значительную роль в рейтинговой технологии обучения. Государственным стандартом предусматривается, как правило, не менее 50% часов из общей трудоемкости дисциплины на самостоятельную работу обучающихся (далее СРО). В связи с этим, обучение включает в себя две, практически одинаковые по объему и взаимовлиянию части – процесса обучения и процесса самообучения. Поэтому СРО должна стать эффективной и целенаправленной работой студента.

Самостоятельная работа обучающихся является одной из основных форм внеаудиторной работы при реализации учебных планов и программ.

Самостоятельная работа – это познавательная учебная деятельность, когда последовательность мышления ученика, его умственных и практических операций и действий зависит и определяется самим студентом.

Обучающийся в процессе изучения дисциплины должен не только освоить учебную программу, но и приобрести навыки самостоятельной работы. Студенту предоставляется возможность работать во время учебы более самостоятельно, чем учащимся в средней школе. Обучающийся должен уметь планировать и выполнять свою работу.

Целью самостоятельной работы студентов является овладение фундаментальными знаниями, профессиональными умениями и навыками деятельности по профилю, опытом творческой, исследовательской деятельности.

1. Цели и задачи дисциплины

Целями освоения дисциплины «Технологии программирования» является формирование компетенций обучающегося в области профессиональной деятельности и сфере профессиональной деятельности:

Об Связь, информационные и коммуникационные технологии в сфере разработки, внедрения и сопровождения информационных технологий и систем

В рамках освоения образовательной программы высшего образования выпускники готовятся к решению задач профессиональной деятельности следующих типов:

- производственно-технологический;
- научно-исследовательский.

2. Методические указания по изучению дисциплины «Технологии программирования»

Основными формами обучения дисциплине являются:

- 1) лекции,
- 2) лабораторные занятия,
- 3) самостоятельная работа.

2.1. Методические рекомендации по подготовке к лекциям

Лекция – логическое изложение материала в соответствии с планом лекции, который сообщается в начале каждой лекции, и имеет законченную форму, т.е. содержит пункты, позволяющие охватить весь материал, который необходимо довести до студентов.

Главной задачей лектора является организация процесса познания студентами материала изучаемой дисциплины на всех этапах ее освоения, предусмотренных федеральным государственным образовательным стандартом.

На лекциях особое внимание уделяется не только усвоению изучаемых проблем, но и стимулированию Вашей активной познавательной деятельности, творческого мышления, развитию научного мировоззрения, профессионально-значимых свойств и

качеств. Лекции по учебной дисциплине проводятся, как правило, как проблемные в форме диалога (интерактивные).

Излагаемый материал может показаться Вам сложным, поскольку включает знания, почерпнутые преподавателем из различных отраслей психологии – общей психологии, психологии познавательных процессов, психологии личности, социальной психологии и т.д. Вот почему необходимо добросовестно и упорно работать на лекциях. Осуществляя учебные действия на лекционных занятиях, Вы должны внимательно воспринимать действия преподавателя, запоминать складывающиеся образы, мыслить, добиваться понимания изучаемого предмета, применения знаний на практике, при решении учебно-профессиональных задач. В ходе лекционных занятий необходимо вести конспектирование учебного материала, обращая внимание на категории, формулировки, раскрывающие содержание тех или иных явлений и процессов, научные выводы и практические рекомендации.

Правила конспектирования:

1. Конспектирование лекций ведется в специально отведенной для этого тетради, каждый лист которой должен иметь поля (4-5 см) для дополнительных записей.

2. Необходимо записывать тему и план лекций, рекомендуемую литературу к теме. Записи разделов лекции должны иметь заголовки, подзаголовки, красные строки. Для выделения разделов, выводов, определений, основных идей можно использовать цветные карандаши и фломастеры.

3. Названные в лекции ссылки на первоисточники надо пометить на полях, чтобы при самостоятельной работе найти и вписать их.

4. В конспекте дословно записываются определения понятий, категорий и законов. Остальное должно быть записано своими словами.

5. Каждому студенту необходимо выработать и использовать допустимые сокращения наиболее распространенных терминов и понятий. Однако чрезмерное увлечение сокращениями может привести к тому, что со временем в них будет трудно разобраться.

6. В конспект следует заносить всё, что преподаватель пишет на доске, а также рекомендуемые схемы, таблицы, диаграммы и т.д. Надо иметь в виду, что изучение и отработка прослушанных лекций без промедления значительно экономит время и способствует лучшему усвоению материала.

Перед очередной лекцией необходимо просмотреть по конспекту материал предыдущей лекции. При затруднениях в восприятии материала следует обратиться к основным литературным источникам. Если разобраться в материале опять не удалось, то обратитесь к лектору (по графику его консультаций) или к преподавателю на лабораторных занятиях.

Содержание разделов учебной дисциплины «Технологии программирования»

1. Введение в технологию программирования.

Введение в технологию программирования, программную инженерию. Понятие программного средства. Технология программирования и основные этапы ее развития. Проблемы разработки сложных программных систем. Жизненный цикл программного средства. Архитектура ПО. Системный анализ при создании ПС.

2. Организация процесса проектирования программного обеспечения.

Системный подход при разработке ПС. Модели разработки: каскадная, с промежуточным контролем, спиральная и т.д.; CASE-технологии. Тестирование и оценка качества. Управление проектом, планирование и распределение ресурсов, контроль исполнения сроков.

3. Методы проектирования программного обеспечения

Использование декомпозиции и абстракции при проектировании ПО. Спецификация процедур и данных. Внешняя и внутренняя спецификации. Декомпозиция задачи. Методы проектирования структуры ПО.

4. Парадигмы программирования.

Парадигмы программирования: визуальная, функциональная, процедурная, объектно-ориентированная. Объектно-ориентированная парадигма: понятия объекта, класса объектов; основные принципы объектно-ориентированного программирования (абстракция, инкапсуляция, наследование и полиморфизм); классы и объекты; интерфейсы и реализация.

5. Технология создания программного кода.

Библиотеки стандартных компонентов, библиотеки объектов. Многоокопные интерфейсы; примеры реализации интерфейсов с пользователем с использованием графических пакетов. Статические и динамические типы данных. Простые и составные типы данных

6. Технологические средства разработки программного обеспечения.

Инструментальная среда разработки. Средства поддержки проекта. Отладчики. CASE-технология. Язык UML.

7. Методы отладки и тестирования программ.

Категории программных ошибок. Типы тестов. Тестирование на этапе планирования. Тестирование на этапе проектирования. Тестирование на стадии кодирования. Регрессионное тестирование. Разработка тестов.

8. Документирование программных средств.

Документация, создаваемая и используемая в процессе разработки программных средств. Пользовательская документация программных средств. Документация по сопровождению программных средств.

2.2. Методические рекомендации по подготовке к лабораторным занятиям

Семинар – это один из наиболее сложных и в то же время плодотворных видов (форм) вузовского обучения и воспитания. В условиях высшей школы Лабораторная работа – вид практической работы, проводимой под руководством преподавателя, ведущего научные исследования по тематике лабораторной работы и в данной отрасли научного знания.

Лабораторная работа предназначен: для углубленного изучения той или иной дисциплины и овладения методологией применительно к особенностям изучаемой отрасли науки; для активной самостоятельной групповой работы, когда студенты могут подготовить, обдумать поставленные перед ними проблемы, проверить свою позицию, услышать и обсудить другие.

Целесообразно готовиться к лабораторной работе занятиям за 1- 2 недели до их начала. Начинать надо с изучения рекомендованной литературы, так как на лекции обычно

рассматривается не весь материал, а только его часть. Остальная его часть восполняется в процессе самостоятельной работы. Особое внимание при этом необходимо обратить на содержание основных положений и выводов, объяснение явлений и фактов, уяснение практического приложения рассматриваемых теоретических вопросов. В процессе этой работы вы должны стремиться понять и запомнить основные положения рассматриваемого материала, примеры, поясняющие его, а также разобраться в иллюстративном материале.

Заканчивать подготовку следует составлением плана (конспекта) по изучаемому материалу (вопросу). Это позволяет составить концентрированное, сжатое представление по изучаемым вопросам.

На лабораторной работе каждый из Вас должен быть готовым к выступлению по всем поставленным в плане вопросам, проявлять максимальную активность при их рассмотрении. Выступление должно строиться свободно, убедительно и аргументировано. Необходимо, чтобы выступающий проявлял собственное отношение к тому, о чем он говорит, высказывал свое личное мнение, понимание, обосновывал его и мог сделать правильные выводы из сказанного. При этом Вы можете обращаться к записям конспекта и лекций, непосредственно к первоисточникам, использовать знание художественной литературы и искусства, факты и наблюдения современной жизни и т.д. Вокруг такого выступления могут разгореться споры, дискуссии, к участию в которых должен стремиться каждый.

При подготовке к лабораторной работе вам следует:

- приносить с собой рекомендованную преподавателем литературу к конкретному занятию;
- до очередного практического занятия по рекомендованным литературным источникам проработать теоретический материал, соответствующей темы занятия;
- при подготовке к лабораторной работе следует обязательно использовать не только лекции, но учебную, методическую литературу;
- в начале занятий задать преподавателю вопросы по

материалу, вызвавшему затруднения в его понимании и освоении при решении задач, заданных для самостоятельного решения;

- в ходе лабораторной работы давать конкретные, четкие ответы по существу вопросов;

- на занятии демонстрировать понимание проведенных анализов, ситуаций, в случае затруднений обращаться к преподавателю.

Если Вы пропустили занятие (независимо от причин) или не подготовились к занятию, рекомендуется не позже, чем в 2-недельный срок явиться на консультацию к преподавателю и отчитаться по теме, изученной на занятии. Студенты, не отчитавшиеся по каждой не проработанной ими на занятиях теме к началу зачетной сессии, упускают возможность получить положительную оценку в соответствующем семестре. При такой подготовке лабораторное занятие пройдет на необходимом методологическом уровне и принесет интеллектуальное удовлетворение всей группе.

Темы лабораторных работ

1. Технологический цикл решения задачи на ЭВМ.
2. Средства отладки и обработка ошибок.
3. Декомпозиция задачи. Методы проектирования структуры ПО.
4. Построение и программирование основных алгоритмических конструкций.
5. Построение и программирование основных алгоритмических конструкций при работе с массивами
6. Программирование структур и объединений
7. Одно и двунаправленные списки, стеки, очереди.
8. Разработка программ с применением обобщенной (шаблоны, STL) и событийной (Win 32 API и MFC) технологии программирования.
9. Подключение к базе данных. Разработка проекта взаимодействующего с базой данных через SQL-команды для хранения, отображения, редактирования, добавления и удаления информации.

10.Создание пользовательского интерфейса, меню. Откладка и тестирование программных проектов из предыдущих практических занятий. Устранение их логических (алгоритмических) ошибок.

3. Методические рекомендации по организации самостоятельной работы обучающихся по дисциплине «Технологии программирования»

3.1. Общие методические рекомендации по самостоятельной работе

Самостоятельная работа – это планируемая работа студентов, выполняемая по заданию и при методическом руководстве преподавателя, но без его непосредственного участия. Целью самостоятельной работы студентов являются: обучение навыкам работы с научной литературой и практическими материалами, необходимыми для углубленного изучения дисциплины, а также развитие у них устойчивых способностей к самостоятельному (без помощи преподавателя) изучению и изложению полученной информации. В связи с этим основными задачами самостоятельной работы студентов, изучающих дисциплину являются:

– во-первых, продолжение изучения учебной дисциплины в домашних условиях по программе, предложенной преподавателем;

– во-вторых, привитие студентам интереса к психологической литературе;

– в-третьих, развитие познавательных способностей.

Изучение и изложение информации, полученной в результате изучения научной литературы и практических материалов, предполагают развитие у студентов как владения навыками устной речи, так и способностей к четкому письменному изложению материала.

Основными формами самостоятельной работы студентов являются:

- подготовку к аудиторным занятиям, изучение материала по учебникам (в т.ч. по конспекту лекций);

- оформление отчетов по лабораторным работам

(подготовка к лабораторным занятиям);

- выполнение курсовой работы.

Основной формой контроля за самостоятельной работой студентов являются лабораторные занятия, промежуточная аттестация, а также еженедельные консультации преподавателя по выполнению курсовой работы.

3.2. Методические рекомендации по курсовой работе

Курсовая работа - это один из видов самостоятельной работы студентов, основной задачей которого является выработка у студентов навыков архивного исследования, умения формулировать ведущую проблему, анализировать источники и литературу по данной проблематике, давать интерпретацию знаний по теории информации, источников, показать владение необходимым математическим инструментарием.

Курсовая работа призвана показать следующие знания студента:

- основные понятия и термины теории информации, теоретические положения, современные методы и средства получения, представления, обработки и хранения информации;
- знание основной литературы и Интернет-источников по выбранной теме.

В курсовой работе определяются приобретаемые компетенции, самостоятельность формулирования задачи по выбранной теме, способность ее решения с помощью собственных знаний и знаний, полученных из открытых источников, возможности оценки получаемых результатов и применения их на практике, которые являются показателями профессиональной подготовки студента.

Курсовая работа должна быть объёмом от 15 до 25 листов в печатном виде.

Выбор темы курсовой работы осуществляется по желанию студента из приведенных тем в первом параграфе.

Грамотно сформулированная тема зафиксировала предмет изучения; задача студента — найти информацию, относящуюся к данному предмету и разрешить поставленную проблему.

Основными процедурами выполняемой курсовой работы являются:

- формализация постановки решаемой задачи;
- анализ и уточнение требований к разрабатываемому программному обеспечению для автоматизации информационных процессов в предметной области;
- выбор структур данных, методов и алгоритмов для автоматизации информационных процессов в предметной области;
- разработка схемы данных программного средства автоматизации информационных процессов в предметной области;
- проектирование интерфейса пользователя (форм и отчетов) разрабатываемого приложения для автоматизации информационных процессов в предметной области;
- выбор стратегии тестирования и разработка набора тестов программного средства автоматизации информационных процессов в предметной области.

Выполнение курсовой работы предусмотрено учебным планом и обязательно для каждого студента. В результате выполнения курсовой работы студент должен показать готовность к владению основными умениями вести исследовательскую деятельность, продемонстрировать необходимые компетенции, приобрести новые знания, умения и получить опыт. С этой целью будущим специалистам необходимо:

- научиться пользоваться библиографическими указателями в научной библиотеке университета источников обеспечивающих дисциплину, по которой выполняется курсовая работа получить необходимый минимум знаний по теме и уметь зафиксировать нужную информацию;
- грамотно изложить состояние изучаемого вопроса на основе анализа современной литературы;
- выполнить проектирование и реализацию приложения по выбранной теме, проанализировать результат и сделать

выводы.

Работу над курсовой работой следует начинать с подбора литературы по теме исследования. При этом выделяются следующие источники библиографической информации:

- библиотечные фонды, которые включают:
- первичные (статьи, диссертации, монографии и т. д.);
- вторичные (библиография, реферативные журналы, сигнальная информация и т.д.);
- интернет-ресурсы, которые состоят из обзоров, компилятивных работ, справочных данных, электронных учебников, видео уроков, форумов, иностранных статей и книг, которые можно перевести с помощью электронных переводчиков):
- лекционный материал по поддерживаемой курсовой работой дисциплине.

В результате сбора и сравнительного анализа информации по теме исследования должны быть получены следующие сведения:

- какие наивысшие достижения в этой области имеются на текущий день;
- какие специалисты и ученые наиболее авторитетны и в каких научных учреждениях, выполняются работы по теме исследования;
- наиболее значимые результаты такой работы и публикации (в каких источниках);
- в чем конкретно они состоят?

Главным результатом должно быть заключение на основании сравнительного анализа по выбранным критериям достоинств и недостатков методов и средств решений поставленной задачи в курсовой работе и особенный упор необходимо сделать на тех сторонах, где найденные решения недостаточно проработаны или не завершены. Именно они должны войти в формулировку основной задачи курсовой работы и послужить базой для поиска эффективных средств ее решения.

Курсовая работа завершается заключением, в котором

указываются основные результаты, их важность для науки и техники, общественная значимость, дальнейшие перспективы.

Для написания обзора по теме исследования целесообразно использовать не менее 20 источников. Следует отметить, что в тексте работы на все приведенные источники используемой литературы должны иметься обязательные ссылки.

Рекомендуемые темы курсовой работы

1. Теоретическая часть. Объектно-ориентированный подход к программированию и технология .NET /ASP.NET (C++, C#, Python).

2. Практическая часть.

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».

Критерии оценивания:

- полнота и правильность ответа;
- степень осознанности, понимания изученного;
- языковое оформление ответа.

Показатели и шкала оценивания выполнения курсовой работы

Шкала оценивания	Показатели
5	работа выполнена без ошибок, обучающийся представил оригинальное и грамотное решение, четко и грамотно оформляет пояснительную записку без отступлений от требований к её оформлению, подробно и безошибочно отвечает на все заданные ему вопросы, проявляет при работе достаточную самостоятельность
4	работа выполнена с незначительными ошибками, но при опросе обучающийся проявляет понимание ошибок и способов их исправления, не допускает существенных погрешностей в ответах на вопросы, аккуратно выполняет демонстрационный материал и пояснительную записку
3	работа выполнена без грубых ошибок, но при опросе обучающийся проявляет недостаточное понимание всех подробностей проделанной работы; допускает при ответах на вопросы неточности и неправильные

	формулировки; допускает небрежность в графической работе и в оформлении пояснительной записки.
2	принципиальные ошибки в представленной к защите работе и обучающийся при ответах на вопросы, не может устранить указанные недостатки, небрежно выполняет работу и представляет неполную и не соответствующую правилам оформления пояснительную записку, проявляет полное пренебрежение к срокам выполнения проекта.

4. Промежуточная аттестация

Итоговой оценкой по дисциплине является результат промежуточной аттестации, выставленный с учетом результатов текущего контроля.

Примерные тест к экзамену

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.

Показатели и шкала оценивания тестовых заданий на экзамене

Текущая аттестация	Количество баллов	Шкала оценивания
выполнение требований по текущей аттестации в полном объеме	90% - 100%	5
	80% - 89%	4
выполнение требований по текущей аттестации в неполном объеме	60% - 79%	3
невыполнение требований по текущей аттестации	менее 60%	2

5. Перечень основной, дополнительной учебной литературы и учебно-методической литературы для самостоятельной работы обучающихся, необходимой для освоения дисциплины

Основная литература

Гниденко, И. Г. Технологии и методы программирования : учебное пособие для вузов / *И. Г. Гниденко, Ф. Ф. Павлов,*

Д. Ю. Федоров. — Москва : Издательство Юрайт, 2023. — 235 с. — (Высшее образование). — ISBN 978-5-534-02816-4. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/511891>

Малов, А. В. Концепции современного программирования : учебное пособие для вузов / А. В. Малов, С. В. Родионов. — Москва : Издательство Юрайт, 2023. — 96 с. — (Высшее образование). — ISBN 978-5-534-14911-1. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/520338>

Дополнительная литература

Подбельский, В. В. Программирование. Базовый курс С# : учебник для вузов / В. В. Подбельский. — Москва : Издательство Юрайт, 2023. — 369 с. — (Высшее образование). — ISBN 978-5-534-10616-9. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/511747>

Огнева, М. В. Программирование на языке С++: практический курс : учебное пособие для вузов / М. В. Огнева, Е. В. Кудрина. — Москва : Издательство Юрайт, 2023. — 335 с. — (Высшее образование). — ISBN 978-5-534-05123-0. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/515142>

Зыков, С. В. Программирование : учебник и практикум для академического бакалавриата / С. В. Зыков. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2023. — 285 с. — (Бакалавр. Академический курс). — ISBN 978-5-534-16031-4. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/530294>

Зыков, С. В. Объектно-ориентированное программирование : учебник и практикум для вузов / С. В. Зыков. — 2-е изд. — Москва : Издательство Юрайт, 2023. — 151 с. — (Высшее образование). — ISBN 978-5-534-16941-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/532054>



Издается в авторской редакции
Подписано в печать 29.06.2023. Формат 60x90 ¹/₁₆
Бумага кн.-журн. П.л. 1,75 Гарнитура Таймс.
Тираж 30 экз.

Воронежский филиал Федерального государственного образовательного
учреждения высшего образования
«Государственный университет морского и речного флота имени
адмирала С.О. Макарова»
Типография Воронежского филиала ФГБОУ ВО «ГУМРФ имени
адмирала С.О. Макарова», Воронеж, Ленинский проспект, 174л.

Отпечатано с оригинал-макета заказчика. Ответственность за содержание
представленного оригинал-макета типография не несет.
Требования и пожелания направлять авторам данного издания.