

**НЕГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«САНКТ-ПЕТЕРБУРГСКИЙ
ГУМАНИТАРНЫЙ УНИВЕРСИТЕТ ПРОФСОЮЗОВ»**

Кафедра Информатики и математики
(полное наименование кафедры)

УТВЕРЖДЕН
на заседании кафедры

Протокол №1 от 01.06.2020

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ПО ДИСЦИПЛИНЕ

Программная инженерия

(наименование дисциплины)

09.03.03 «Прикладная информатика»

(код наименования направления подготовки /специальности/)

Прикладная информатика в экономике

(направленность/профиль/)

Санкт-Петербург

1. Общие положения

Фонд оценочных средств (ФОС) по дисциплине используется в целях нормирования процедуры оценивания качества подготовки и осуществляет установление соответствия учебных достижений запланированным результатам обучения и требованиям образовательной программы дисциплины. Предметом оценивания являются знания, умения, навыки и (или) опыт деятельности, характеризующие этапы формирования компетенций у обучающихся. Процедуры оценивания применяются в процессе обучения на каждом этапе формирования компетенций посредством определения для отдельных составных частей дисциплины методов контроля – оценочных средств. Основным механизмом оценки качества подготовки и формой контроля учебной работы студентов являются текущий контроль успеваемости и промежуточная аттестация.

1.1. Цель и задачи текущего контроля студентов по дисциплине

Цель текущего контроля – систематическая проверка степени освоения программы 09.03.03 «Прикладная информатика» дисциплины «Программная инженерия» уровня достижения планируемых результатов обучения - знаний, умений, навыков, в ходе ее изучения при проведении занятий, предусмотренных учебным планом. Задачи текущего контроля:

1. обнаружение и устранение пробелов в освоении учебной дисциплины;
2. своевременное выполнение корректирующих действий по содержанию и организации процесса обучения;
3. определение индивидуального учебного рейтинга студентов;
4. подготовка к промежуточной аттестации.

В течение семестра при изучении дисциплины реализуется традиционная система поэтапного оценивания уровня освоения. За каждый вид учебных действий студенты получают оценку.

1.2. Цель и задачи промежуточной аттестации студентов по дисциплине.

Цель промежуточной аттестации – проверка степени усвоения студентами учебного материала, уровня достижения планируемых результатов обучения и сформированности компетенций на момент завершения изучения дисциплины. Промежуточная аттестация проходит в форме зачёта.

Задачи промежуточной аттестации:

1. определение уровня освоения учебной дисциплины;
2. определение уровня достижения планируемых результатов обучения и сформированности компетенций;
3. соотнесение планируемых результатов обучения с планируемыми результатами освоения образовательной программы в рамках изученной дисциплины.

				<p>Задания для самостоятельной работы</p> <p>Задания для самостоятельной работы</p>
2	Анализ и проектирование программного обеспечения	ОПК-7	<p>ОПК-7.1. Знает принципы сбора, отбора и обобщения информации.</p> <p>ОПК-7.2. Умеет проводить анализ и проектирование современного программного обеспечения, определять его качественные показатели, организовывать процесс разработки и вести документацию в соответствии с современными стандартами.</p> <p>ОПК-7.3. Владеет навыками программирования, отладки и тестирования программного обеспечения.</p> <p>ОПК-9.1. Знает принципы организации проектирования и содержание этапов процесса разработки программных комплексов дисциплины, задачи и методы исследования и обеспечения качества и надежности программных компонентов, основы разработки программных комплексов.</p> <p>ОПК-9.2. Умеет проводить анализ и проектирование современного программного обеспечения, определять его качественные показате-</p>	<p>Опрос</p> <p>Задания для самостоятельной работы</p> <p>Задания для самостоятельной работы</p>

		ОПК-9	<p>программного обеспечения, определять его качественные показатели, организовывать процесс разработки и вести документацию в соответствии с современными стандартами.</p> <p>ОПК-7.3. Владеет навыками программирования, отладки и тестирования программного обеспечения.</p> <p>ОПК-9.1. Знает принципы организации проектирования и содержание этапов процесса разработки программных комплексов дисциплины, задачи и методы исследования и обеспечения качества и надежности программных компонентов, основы разработки программных комплексов.</p> <p>ОПК-9.2. Умеет проводить анализ и проектирование современного программного обеспечения, определять его качественные показатели, организовывать процесс разработки и вести документацию в соответствии с современными стандартами.</p> <p>ОПК-9.3. Владеет навыками работы с инструментальными средствами моделирования предметной области, прикладных и информационных процессов, использования технологических стандартов информационных систем и программных продуктов.</p>	<p>Задания для самостоятельной работы</p> <p>Задания для самостоятельной работы</p> <p>Опрос</p> <p>Задания для самостоятельной работы</p>
--	--	-------	--	--

				Задания для самостоятельной работы
4	Стандартизация и сертификация процесса разработки информационных технологий и программного обеспечения	ОПК-7	<p>ОПК-7.1. Знает принципы сбора, отбора и обобщения информации.</p> <p>ОПК-7.2. Умеет проводить анализ и проектирование современного программного обеспечения, определять его качественные показатели, организовывать процесс разработки и вести документацию в соответствии с современными стандартами.</p> <p>ОПК-7.3. Владеет навыками программирования, отладки и тестирования программного обеспечения.</p> <p>ОПК-9.1. Знает принципы организации проектирования и содержание этапов процесса разработки программных комплексов дисциплины, задачи и методы исследования и обеспечения качества и надежности программных компонентов, основы разработки программных комплексов.</p> <p>ОПК-9.2. Умеет проводить анализ и проектирование современного программного обеспечения, определять его качественные показатели, организовывать процесс разработки и вести документацию в соответствии с современными стандартами.</p> <p>ОПК-9.3. Владеет навыками работы с инструментальными средствами моделирования предметной области, прикладных и информационных процессов, использования технологических стандартов информационных систем и программных продуктов.</p>	<p>Опрос</p> <p>Задания для самостоятельной работы</p> <p>Задания для самостоятельной работы</p> <p>Опрос</p>

				Задания для самостоятельной работы
				Задания для самостоятельной работы
Результат достижения планируемых результатов изучения дисциплины				зачёт

3. Описание показателей и критериев оценивания компетенций

3.1. Критерии оценивания (текущий контроль)

1. Оценка «отлично» выставляется студенту, если студент имеет глубокие знания учебного материала по теме практического задания, в логической последовательности излагает материал; смог ответить на все уточняющие и дополнительные вопросы;
2. Оценка «хорошо» выставляется, если студент показал знание учебного материала, смог ответить почти полностью на все заданные дополнительные и уточняющие вопросы;
3. Оценка «удовлетворительно» выставляется, если студент в целом освоил материал; однако, ответил не на все уточняющие и дополнительные вопросы;
4. Оценка «неудовлетворительно» выставляется студенту, если он имеет существенные пробелы в знаниях основного учебного материала по теме практического задания, который полностью не раскрыл содержание вопросов, не смог ответить на уточняющие и дополнительные вопросы.

3.2. Критерии оценивания (зачет)

Знания, умения, навыки и компетенции студентов оцениваются следующими оценками: «зачтено», «не зачтено».

«Зачтено» выставляется студенту при условии, что студент твердо знает программный материал, грамотно и последовательно его излагает, увязывает с практикой, владеет необходимыми умениями и навыками в выполнении практических заданий и решении задач.

«Не зачтено» выставляется студенту при условии, что студент не знает значительной части основного программного материала, в ответе допускает существенные ошибки, неправильные формулировки, не владеет необходимыми умениями и навыками в выполнении практических заданий и решении задач.

ПРОМЕЖУТОЧНАЯ АТТЕСТАЦИЯ

Вопросы для подготовки к промежуточной аттестации по дисциплине (зачету)

1. Жизненный цикл программного обеспечения. Основные стадии и рабочие продукты.
2. Каскадная модель жизненного цикла ПО и ее недостатки. Каскадная модель с перекрытиями.
3. Сравнение итерационной и контролируемой итерационной моделей.
4. Спиральная модель. Роль прототипирования в этой модели. Прототипы и их классификация.
5. Стадии и рабочие процессы жизненного цикла. Организационные мероприятия.
6. Программное обеспечение поддержки жизненного цикла.
7. Классификация CASE. Понятия методологии, метода, нотации и средства CASE.
8. Репозиторий и артефакты процесса разработки.
9. Возвратное проектирование.
10. Реинжиниринг. Стадии обратного и прямого инжиниринга. Редокументирование.
11. Обзор каталогов шаблонов проектирования и рефакторингов.
12. Анализ и проектирование программного обеспечения на основе структурного подхода.
13. Диаграммы потоков данных.
14. Моделирование данных.
15. Диаграммы Костантайна.
16. Методология ОМТ.
17. Методика Шлеера-Меллора.
18. Унифицированный язык моделирования UML. Элементы моделей, основные отношения. Механизмы расширения UML.
19. Статические диаграммы. Классов, объектов, пакетов.
20. Динамические диаграммы. Диаграммы действий. Диаграммы взаимодействия. Диаграммы состояний.
21. Диаграммы для физического представления системы. Диаграммы компонентов и диаграммы размещения.
22. Метрики в проектах: метрики процессов и улучшение процессов программного обеспечения, метрики проектов. Роль менеджера в измерении и анализе метрик.
23. Программное обеспечение для измерения и контроля метрик.
24. Меры Холстеда - анализ применимости.
25. Сравнение топологических мер сложности.
26. Меры сложности, основанные на концепции информационных потоков между компонентами системы.
27. Особенности архитектурных мер для анализа многомодульных программ.

28. Обзор объектно-ориентированных метрик.
29. Показатели качества. Характеристики качества и их влияние на различных этапах жизненного цикла программного продукта.
30. Международные, национальные, отраслевые стандарты и организации.
31. Роль и место стандартов в процессе разработки программного обеспечения.
32. Стандарты серии ISO 9000.
33. ГОСТ “Качество программных средств”.
34. ГОСТ “Оценка качества программных средств”.
35. Модель зрелости процесса разработки программного обеспечения СММ. Уровни зрелости. Ключевые области обследования.
36. Унифицированный процесс разработки объектно-ориентированного программного обеспечения с использованием UML (RUP).
37. Рабочие процессы, работы, роли, артефакты и шаблоны RUP.
38. Рабочие процессы для малых проектов. Экстремальное программирование (XP).

4. Типовые контрольные задания (тесты, рефераты, курсовые работы, кейсы и др.) и методические материалы, процедуры оценивания знаний, умений и навыков

Методические рекомендации по написанию контрольных работ

Важнейшей формой учебной отчетности студента является **контрольная работа**.

Выполнение контрольной работы является промежуточной формой отчетности по изучаемой дисциплине и преследует цель лишь оценить способность студента к самостоятельному поиску источников, формированию содержания и его письменного изложения по указанной проблеме. Это важная составляющая изучения дисциплины, а также эффективная форма контроля знаний. При заочном обучении она выступает как обязательная, основная форма самостоятельной работы. В работе (в соответствии с учебным планом) студент обязан самостоятельно глубоко разобраться в изучаемых проблемах, усвоить суть темы, уяснить ее содержание и только затем письменно представить свою отчетную работу.

Выполнение контрольной работы является одним из условий допуска студента к сдаче экзамена. Работа должна соответствовать установленным требованиям, то есть в ней должны быть раскрыты все проблемы, определенные темой. Для этого студент обязан самостоятельно проанализировать первоисточники и дать исчерпывающие ответы на вопросы темы. Контрольная работа — серьезное учебное задание, и чтобы написать ее как следует, необходимо использовать те первоисточники и учебные пособия, которые позволяют полнее разобраться в проблеме. Студент должен регулярно работать в университетской и городской библиотеке, вдумчиво конспектировать лекции преподавателей.

При написании контрольной работы следует обращать особое внимание на грамотное использование терминологии. При употреблении впервые тех или иных терминов и понятий следует давать их определения либо в самом тексте, либо в сносках.

Приступая к контрольной работе, требуется сначала ознакомиться с имеющейся литературой по теме, изучить первоисточники и составить план. Здесь, в отличие от курсовой работы, план предполагает рассмотрение одной, причем довольно широкой, проблемы, и он может состоять из двух-трех вопросов. Минимальное количество первоисточников, привлекаемых для написания курсовой работы — пять наименований.

Как правило, контрольные работы по дисциплине сугубо индивидуальны, то есть их тематика персонифицирована. Однако в отдельных случаях темы контрольных работ могут быть адресованы и сразу нескольким, и группе в целом. Таким приемом преподаватель выявляет степень усвоения какой-то важной учебной проблемы и определяет необходимость про-

ведения дополнительных занятий по какой-либо теме. В настоящее время широко используется методика компьютерного тестирования знаний студентов по дисциплинам, в результате чего появляется возможность быстро проверять знания по наиболее важным темам и объективно оценивать их. Эта форма также может выступать как вид контрольной работы.

В качестве контрольной работы широко применяется самостоятельное изучение монографического исследования по конкретной, крайне важной проблеме, требующей глубокого рассмотрения. Этот вид работы предполагает не простое знакомство с определенным монографическим исследованием, а детальное его изучение. Для этого студенту важно знать некоторые правила работы с первоисточником, которым для него будет являться монография. Следует выяснить фамилию автора, его имя и отчество, ученую степень и звание, а также что побудило его взяться за изучение данной проблемы; обратить внимание на основные вопросы монографии и их разрешение автором, уметь раскрывать их в ходе собеседования с преподавателем.

Студенту следует письменно (предельно кратко) очертить те вопросы (полностью или частично), которые поставлены автором в монографическом исследовании; при изложении их следует указывать страницы источника.

Задания для написания контрольных работ (для заочной формы обучения)

Целями выполнения контрольных работ по учебной дисциплине являются расширение и углубление знаний по определенным темам курса, а также демонстрация навыков решения теоретических и практических проблем. Контрольная работа содействует развитию творческого отношения к учебной деятельности, повышает ее эффективность, а также активизирует усилия студента по освоению изучаемого предмета.

Дисциплина преподается студентам 2 курса в третьем семестре, она является логическим продолжением дисциплины «Информатика и программирование» и обеспечивает знания и навыки в области разработки и стандартизации программного обеспечения и информационных технологий.

Методические рекомендации содержат перечень вариантов заданий для выполнения контрольной работы по дисциплине. Каждый вариант содержит теоретический вопрос и практическое задание. Ответ на вопрос оформляется в форме мини реферата объемом не более 5 страниц. Результаты выполнения задания представляются соответствующими иллюстрациями, расчетами и формулами. Выбор варианта задания осуществляется преподавателем.

Вариант №1

- *Перечислите модели жизненного цикла программного обеспечения.*
- *Практическое задание:*

Требуется разработать с использованием UML модель сценариев взаимодействия (**диаграмма прецедентов использования и диаграммы последовательности**) с программным обеспечением банкомата. В его состав банкомата входят следующие устройства:

- дисплей;
- панель управления с кнопками;
- приемник кредитных карт;
- хранилище денег и лоток для их выдачи;
- принтер для печати справок.

Банкомат подключен к линии связи для обмена данными с компьютером банка, хранящим сведения о счетах клиентов. После распознавания типа пластиковой карточки, банкомат выдает на дисплей приглашение ввести персональный код. Затем банкомат проверяет правильность введенного кода и предлагает выбрать операцию. Клиент может либо снять наличные со счета, либо узнать остаток на его счету. После выбора клиентом суммы банкомат спрашивает, нужно ли печатать справку по операции. Затем банкомат посылает запрос на снятие выбранной суммы компьютеру банка. В случае получения разрешения на операцию, банкомат проверяет, имеется ли требуемая сумма в его хранилище денег. Если он может выдать деньги, то на дисплей выводится сообщение «Выньте карту». После удаления карточки из

приемника, банкомат выдает указанную сумму в лоток выдачи. Затем банкомат печатает справку по произведенной операции, если она была затребована клиентом.

Вариант №2

- *Какие программы для поддержки жизненного цикла программного обеспечения Вы знаете?*
- *Практическое задание:*

Требуется разработать с использованием UML информационную модель (**диаграммы пакетов и классов**) программного обеспечения банкомата. Банкомат - это автомат для выдачи наличных денег по кредитным пластиковым карточкам. В его состав входят следующие устройства:

- дисплей;
- панель управления с кнопками;
- приемник кредитных карт;
- хранилище денег и лоток для их выдачи;
- принтер для печати справок.

Банкомат подключен к линии связи для обмена данных с компьютером банка, хранящим сведения о счетах клиентов. После распознавания типа пластиковой карточки, банкомат выдает на дисплей приглашение ввести персональный код. Затем банкомат проверяет правильность введенного кода и предлагает выбрать операцию. Клиент может либо снять наличные со счета, либо узнать остаток на его счету. После выбора клиентом суммы банкомат запрашивает, нужно ли печатать справку по операции. Затем банкомат посылает запрос на снятие выбранной суммы компьютеру банка. В случае получения разрешения на операцию, банкомат проверяет, имеется ли требуемая сумма в его хранилище денег. Если он может выдать деньги, то на дисплей выводится сообщение «Выньте карту». После удаления карточки из приемника, банкомат выдает указанную сумму в лоток выдачи. Затем банкомат печатает справку по произведенной операции, если она была затребована клиентом.

Вариант №3

- *Что такое репозиторий и артефакты процесса разработки, реинжиниринг и рефакторинг программного обеспечения?*
- *Практическое задание:*

Требуется разработать с использованием UML модель системы поддержки заказа и учета товаров в магазина. Для каждого товара фиксируется место хранения (определенная полка), количество товара и его поставщик. Система поддержки заказа и учета товаров должна обеспечивать добавление информации о новом товаре, изменение или удаление информации об имеющемся товаре, хранение (добавление, изменение и удаление) информации о поставщиках, включающей в себя название фирмы, ее адрес и телефон. При помощи системы составляются заказы поставщикам. Каждый заказ может содержать несколько позиций, в каждой позиции указываются наименование товара и его количество в заказе. Система учета по требованию пользователя формирует следующую справочную информацию:

- список всех товаров;
- список товаров, имеющихся в наличии;
- список товаров, количество которых необходимо пополнить;
- список товаров, поставляемых данным поставщиком.

Вариант №4

- *Приведите основные объемные метрики Холстеда.*
- *Практическое задание:*

Требуется разработать с использованием UML модель системы автоматизирующей деятельность библиотеки. Система поддержки управления библиотекой должна обеспечивать добавление информации о читателях в регистрационный список, редактирование этой информации и удаление ее. В регистрационном списке хранятся фамилия, имя и отчество читателя; номер его читательского билета и дата выдачи билета. В каталоге библиотеки хранится информация о книгах: название, список авторов, библиотечный шифр, год и место издания, название издательства, общее количество экземпляров книги в библиотеке и количество эк-

земпляров, доступных в текущий момент. Система обеспечивает поиск книг в каталоге на основании введенного шифра или названия книги. В системе осуществляется регистрация взятых и возвращенных читателем книг. Про каждую выданную книгу хранится запись о том, кому и когда была выдана книга, и когда она будет возвращена. При возврате книги в записи делается соответствующая пометка, а сама запись не удаляется из системы. Система также должна выдавать следующую информацию:

- какие книги были выданы за данный промежуток времени;
- какие книги были возвращены за данный промежуток времени;
- какие книги находятся у данного читателя;
- имеется ли в наличии некоторая книга.

Вариант №5

- Назовите основные подходы к анализу топологической сложности программ.
- Практическое задание:

Требуется разработать с использованием UML модель программного обеспечения Интернет-магазина. Интернет-магазин позволяет делать покупки с доставкой на дом. Клиентам магазина имеют доступ к каталогу продаваемых товаров. В каталоге товары распределены по разделам. О каждом товаре доступна полная информация (название, вес, цена, изображение, дата изготовления и срок годности) Предусмотрена система поиска товаров в каталоге. Заполнение каталога информацией происходит автоматически в начале рабочего дня, информация берется из системы автоматизации торговли. При отборе клиентами товаров поддерживается виртуальная «торговая корзина». Любое наименование товара может быть добавлено в «корзину» или изъято с последующим пересчетом общей стоимости покупки. Текущее содержимое «корзины» постоянно показывается клиенту. По окончании выбора товаров производится оформление заказа и регистрация покупателя. Клиент указывает в регистрационной форме свою фамилию, имя и отчество, адрес доставки заказа и телефон, по которому с ним можно связаться для подтверждения сделанного заказа. Заказы передаются для обработки в систему автоматизации торговли. Проверка наличия товаров на складе и их резервирование Интернет-магазином не производятся.

Вариант №6

- Опишите особенности статических диаграмм UML (классов, объектов, пакетов).
- Практическое задание:

Для заданного варианта программы на языке Паскаль составить таблицу (в MS Excel), содержащую все простые операторы программы и число вхождений каждого из них в программу. Вычислить число простых операторов и общее число всех операторов. Аналогично составить таблицу, содержащую все простые операнды программы и число вхождений каждого из них в программу. Вычислить число простых операндов, и общее число всех операндов. Следует учитывать только вычислительные операторы и исключить операторы описания переменных, а также, операторы, обеспечивающие интерфейс с пользователем и выдачу текстовых сообщений.

Составить формулы для вычисления и рассчитать следующие метрические характеристики (по Холстеду):

- длину программы;
- реальный и потенциальный объем программы;
- теоретическую длину программы;
- уровень качества программирования, интеллектуальное содержание, трудоемкость понимания готовой программы, время программирования.

Анализируемая программа на языке Паскаль:

```
program gun;
uses Crt;
var
    v0, alpha, t,
    dt, x, y, x0, y0,
    vx0, vy0, tc      : Extended;
```

```

    N, i          : Integer;
const
    g = 9.81;
begin
    ClrScr;
    WriteLn('Введите начальную скорость');
    ReadLn(v0);
    WriteLn;
    WriteLn('Введите наклон траектории');
    ReadLn(alpha);
    WriteLn;
    WriteLn('Введите число точек');
    ReadLn(N);
    vx0 := v0 * Cos(alpha);
    vy0 := v0 * Sin(alpha);
    x0 := 0;
    y0 := 0;
    T := 2 * vy0 / g;
    dt := T / (N - 1);
    i := 1;
    tc := 0;
    while i <= N do
    begin
        x := x0 + vx0 * tc;
        y := y0 + vy0 * tc - g * Sqr(tc) / 2;
        WriteLn(x, ', ', y);
        Inc(i);
        tc := tc + dt;
        if i mod 20 = 0 then
        begin
            WriteLn('Нажмите <Enter>');
            ReadLn;
        end;
    end;
    WriteLn('Нажмите <Enter>');
    ReadLn;
end.

```

Вариант №7

- В чем отличия динамических диаграмм UML (последовательности, сотрудничества, состояний, действий).
- Практическое задание:

Для заданного варианта программы на языке Паскаль построить управляющий граф, представляющий вычислительный алгоритм программы. Вычислить следующие топологические меры сложности:

- цикломатическое число Мак-Кейба;
- структурную сложность программы по критерию минимального покрытия дуг;
- структурную сложность программы по критерию базовых маршрутов.

Анализируемая программа на языке Паскаль:

```

program diophantine_equation_2;
var
    x, y, z, w, n: LongInt;
begin
    n := MaxLongint - 63;

```

```

n := Trunc(Sqrt(n));
n:=n-8;
x := 0;
WriteLn('Все целые решения уравнения  $x^3 = y^2 + 63$ ,');
WriteLn('для  $1 \leq y \leq n$ , n, ');
for y := 1 to n do
begin
    z := y*y + 63;
    repeat
        Inc(x);
        w := x*x*x;
    until w >= z;
    if w = z then
        WriteLn('(x, y) = (' , x, ', ', y, ')')
    else Dec(x);
end;
Write('Работа закончена, нажмите <Enter>:');
ReadLn;
end.

```

Вариант №8

- Опишите основные принципы и нотации структурного подхода.
- Практическое задание:

Для заданного варианта программы на языке Паскаль составить таблицу (в MS Excel), содержащую все простые операторы программы и число вхождений каждого из них в программу. Вычислить число простых операторов и общее число всех операторов. Аналогично составить таблицу, содержащую все простые операнды программы и число вхождений каждого из них в программу. Вычислить число простых операндов, и общее число всех операндов. Следует учитывать только вычислительные операторы и исключить операторы описания переменных, а также, операторы, обеспечивающие интерфейс с пользователем и выдачу текстовых сообщений.

Составить формулы для вычисления и рассчитать следующие метрические характеристики (по Холстеду):

- длину программы;
- реальный и потенциальный объем программы;
- теоретическую длину программы;
- интеллектуальное содержание, трудоемкость кодирования, трудоемкость понимания готовой программы, ожидаемое число ошибок в программе.

Анализируемая программа на языке Паскаль:

```

program maximum_search;

const
    max = 10000;

var
    random_array : array[1..max] of Integer;
    max_element, index, index_max : Integer;

begin
    Randomize;
    for index := 1 to max do
        random_array[index] := Random(10000);

    max_element := random_array[1];

```

```

for index := 2 to max do
  if max_element < random_array[index] then
    begin
      max_element := random_array[index];
      index_max := index;
    end;
  WriteLn('Значение ', index_max, '-го, максимального элемента равно ', random_array[index_max]);
  WriteLn('Для завершения работы нажмите <Enter>');
  ReadLn;
end.

```

Вариант №9

- *Перечислите основные элементы и виды отношений, используемые на диаграммах UML.*
- *Практическое задание:*

Для заданного варианта программы на языке Паскаль построить управляющий граф, представляющий вычислительный алгоритм программы. Вычислить следующие топологические меры сложности:

- цикломатическое число Мак-Кейба;
- структурную сложность программы по критерию минимального покрытия дуг;
- структурную сложность программы по критерию базовых маршрутов.

Анализируемая программа на языке Паскаль:

```

program eratosphen1;

uses crt;

const max = 1000;

var
  b, j, k : word;
  flag : array[1..max] of boolean;

begin
  clrscr;
  flag[1] := false;
  for j := 2 to max do
    flag[j] := true;
  b := trunc(sqrt(max));
  k := 0;
  while k <= b do
    begin
      repeat
        inc(k)
      until flag[k];
      j := 2 * k;
      while j <= max do
        begin
          flag[j] := false;
          j := j + k;
        end;
    end;
  for j := 1 to max do
    if flag[j] then

```



```

        write(j:8);
    writeln;
    write('Нажмите <Enter>');
    readln;
end.

```

Вариант №10

- Дайте сравнительную характеристику стандартизованным процессам разработки программного обеспечения *Rational Unified Process (RUP)* и экстремального программирования (*XP*).
- *Практическое задание:*

Для заданного варианта программы на языке Паскаль составить таблицу (в MS Excel), содержащую все простые операторы программы и число вхождений каждого из них в программу. Вычислить число простых операторов и общее число всех операторов. Аналогично составить таблицу, содержащую все простые операнды программы и число вхождений каждого из них в программу. Вычислить число простых операндов, и общее число всех операндов. Следует учитывать только вычислительные операторы и исключить операторы описания переменных, а также, операторы, обеспечивающие интерфейс с пользователем и выдачу текстовых сообщений.

Составить формулы для вычисления и рассчитать следующие метрические характеристики (по Холстеду):

- длину программы;
- реальный и потенциальный объем программы;
- теоретическую длину программы;
- оценку уровня качества программирования, интеллектуальное содержание, трудоемкость кодирования, уровень используемого языка программирования, ожидаемое число ошибок в программе.

Анализируемая программа на языке Паскаль:

```

program unit_roundoff_error;

var
    s, u: real;
    k: word;

begin
    u := 1.0;
    while 1.0 + u > 1.0 do u := 0.1 * u;
    s := u;
    for k := 1 to 3 do
        begin
            while 1.0 + u <= 1.0 do u := u + s;
            u := u - s;
            s := 0.1*s;
        end;
        writeln('Ошибка округления единицы: ', u:14);
        write('Нажмите <Enter> ');
        readln;
    end.

```

Требования к оформлению контрольной работы подробно представлены в Положении о бюро контрольных работ, размещенном на сайте Университета в личном кабинете на странице в Системе поддержки самостоятельной работы студентов **ПОЛОЖЕНИЕ О БЮРО КОНТРОЛЬНЫХ РАБОТ** _ для работ студентов заочной формы обучения.

Тестовые материалы ПАСПОРТ ТЕСТОВЫХ ЗАДАНИЙ

1. Общее количество тестовых заданий в базе – 161.
2. Ограничение времени выполнения теста (в минутах) – 30 минут.
3. Автоматическое перемешивание вопросов в тесте: - да (нет).
4. Случайный порядок ответов в тестовом задании: - да (нет).
5. Критерии оценки результатов тестирования:
 - Неудовлетворительно – 0 – 55% правильных ответов.
 - Удовлетворительно - 55 – 75% правильных ответов.
 - Хорошо – 75 - 90% правильных ответов
 - Отлично – 90% и более правильных ответов

Пример тестовых заданий для текущего контроля представлен ниже:

1. Сформулируйте понятие жизненного цикла программного средства
 - **Последовательность этапов, частных работ и операций, регламентирующих процесс создания и эксплуатации программных средств от подготовки технического задания до окончания эксплуатации**
 - Последовательность этапов, частных работ и операций, регламентирующих процесс создания и эксплуатации программных средств от подготовки технического задания до завершения испытаний ряда версий
2. Дайте определение модели жизненного цикла программного средства
 - Каскадный набор процессов по разработке, эксплуатации и сопровождению программного продукта на протяжении всего его жизненного цикла
 - Структура процессов, охватывающая жизнь системы от установления требований к ней до прекращения испытаний различных версий программного продукта
 - Спиральная структура, состоящая из процессов, работ и задач по разработке и эксплуатации информационной системы
 - **Модель создания и использования программного обеспечения, начиная с момента возникновения необходимости в нем и заканчивая полным выходом из употребления**
3. Объясните смысл каскадной и спиральной модели жизненного цикла программного средства
 - **В каскадной модели каждый этап завершается выпуском полного комплекта документации для того, чтобы работа была продолжена на следующем этапе**
 - **В спиральной модели на этапах анализа и проектирования создаются прототипы (версии), неполное завершение работ на каждом этапе позволяет переходить на следующий этап, не дожидаясь завершения предыдущего**
 - В спиральной модели каждый этап завершается выпуском полного комплекта документации для того, чтобы работа была продолжена на следующем этапе

- В каскадной модели на этапах анализа и проектирования создаются прототипы (версии), неполное завершение работ на каждом этапе позволяет переходить на следующий этап, не дожидаясь завершения предыдущего
4. В чем заключаются главные положительные свойства каскадной модели?
- **На каждом этапе формируется законченный набор проектной документации, отвечающий критериям полноты и согласованности**
 - Выполняемые в логичной последовательности этапы работ позволяют планировать сроки завершения всех работ и соответствующие затраты
 - Неполное завершение работ на каждом этапе позволяет переходить на следующий этап, не дожидаясь завершения предыдущего
 - Показать пользователям работоспособный продукт, тем самым активизируя процесс уточнения и дополнения требований
5. Охарактеризуйте недостатки каскадной модели
- **Запаздывание с получением результатов**
 - Модели автоматизации могут устареть одновременно с их утверждением
 - **Не ясны сроки перехода на следующий этап**
6. В чем заключается основная проблема спиральной модели?
- Запаздывание с получением результатов
 - **Модели автоматизации могут устареть одновременно с их утверждением**
 - Не ясны сроки перехода на следующий этап
7. Какой современный вариант каскадной модели Вы знаете
- Контролируемая каскадная
 - Планируемая спиральная
 - Ограниченная водопадная
 - **Каскадная с перекрытиями**
8. Какой современный вариант итерационной модели Вы знаете
- **Контролируемая итерационная**
 - Планируемая спиральная
 - Ограниченная водопадная
 - С перекрытиями
9. Какая модель жизненного цикла используется в RUP
- Контролируемая каскадная
 - Планируемая спиральная
 - Ограниченная водопадная
 - Каскадная с перекрытиями
 - **Контролируемая итерационная**
10. Кольцевая разработка (round-trip design)
- **Это синхронизация рабочих продуктов и генерируемого исходного кода**
 - Модель жизненного цикла
 - **Может быть релизована с использованием меток в исходной коде**

- Может быть релизована с использованием репозитория
11. Вариант реализации проектируемой системы или ее аспекта, абстрагирующийся от требований к системе называется
- Модель
 - Итерация
 - Артефакт
 - Прототип
 - Образец
12. При построении прототипа могут быть изменены требования к
- Надежности
 - Языку реализации
 - Качеству
 - Назначению
 - Решаемой задаче
 - Интерфейсу
 - Функциональности
13. Рефакторинг – это
- Изменения выполняемые во внутренней структуре ПО без изменения функциональности
 - Изменения выполняемые во внутренней структуре ПО с изменением функциональности
 - Изменение времени разработки с целью масштабирования
14. Почему следует производить рефакторинг?
- Улучшает процесс проектирования ПО
 - Добавляет новые функции
 - Позволяет находить ошибки
 - Делает ПО легче познаваемым
 - Улучшает стиль программирования
 - Сокращает время разработки
15. Когда следует проводить рефакторинг?
- Когда добавляется новая функция
 - Постоянно
 - После тестирования
 - Когда исправляются ошибки
 - При обзоре кода
16. Множество канонических типов диаграмм UML включает в себя
- Диаграммы классов
 - Диаграммы состояний
 - Диаграммы последовательности
 - Диаграммы потоков данных
17. Имя ассоциации
- образуется из имен ассоциированных классов
 - образуется из имен ролей ассоциированных классов
 - указывается в виде текста над линией ассоциации

18. Если множественность роли ассоциации задана символом * , то это означает, что
- ни одного экземпляра классификатора на не участвует в связях, порождаемых ассоциацией
 - по меньшей мере один экземпляр классификатора участвует в связях, порождаемых ассоциацией
 - **неопределенное количество экземпляров классификатора участвует в связях, порождаемых ассоциацией**
19. Что такое KAELOC
- **Тысяч эквивалентных строк кода на Ассемблере**
 - Тысяч эквивалентных строк машинного кода
 - Тысяч строк кода после трансляции
 - Тысяч строк кода языке программирования
20. В метриках Холстеда что больше длина программы или реальный размер
- Длина
 - **Размер**
 - Они равны
21. В метриках Холстеда интеллектуальное содержание программы инвариантно по отношению к используемым языкам реализации
- **Да**
 - Нет
 - Зависит от уровня абстрактности языка программирования
22. Число Страуда
- Для обычных людей находится в интервале от -5 до 10
 - Для обычных людей находится в интервале от 5 до 100
 - Для обычных людей находится в интервале от 0 до 50
 - **Для обычных людей находится в интервале от 5 до 10**
23. В метриках Холстеда формула ожидаемого числа ошибок в программе
- **Получена экспериментально**
 - Доказана теоретически
 - Отсутствует
24. Цикломатическое число Мак-Кейба
- Числу компонентов связности графа в не связном графе
 - **Равно числу линейно независимых контуров в сильносвязном графе**
 - Числу предикатов в условиях компонентов связности графа
25. Критерий минимального покрытия дуг графа требует
- **Минимального множества маршрутов программы, охватывающих все последовательности передач управления и учитывающих исполнение программы по каждому направлению при ветвлении.**
 - Соответствует однократной проверке каждого линейно-независимого ациклического маршрута и каждого линейно-независимого цикла, в совокупности образующих базовые маршруты
26. Для структурированных программ

- Цикломатическое число на 1 меньше количества вершин, в которых происходит ветвление
 - **Цикломатическое число на 1 больше количества вершин, в которых происходит ветвление**
 - Цикломатическое число равно количеству вершин, в которых происходит ветвление
27. В модели зрелости процесса разработки ПО СММ
- **Пять уровней зрелости, но сертификация производится только по трем**
 - Пять уровней зрелости, но сертификация производится только по четырем
 - Четыре уровня зрелости, но сертификация производится только по двум
28. В модели зрелости процесса разработки ПО СММ на втором уровне главная фигура от которой зависят сроки и качество проекта – это
- **Менеджер проекта**
 - Руководитель организации
 - Заказчик
 - Программист
 - Тестировщик
 - Нет правильного ответа
29. В модели зрелости процесса разработки ПО СММ на первом уровне главная фигура от которой зависят сроки и качество проекта – это
- **Менеджер проекта**
 - Руководитель организации
 - Заказчик
 - **Программист**
 - Тестировщик
 - Нет правильного ответа
30. В модели зрелости процесса разработки ПО СММ на третьем уровне главная фигура от которой зависят сроки и качество проекта – это
- Менеджер проекта
 - Руководитель организации
 - Заказчик
 - Программист
 - Тестировщик
 - **Нет правильного ответа**
31. В модели зрелости процесса разработки ПО СММ успешность использования ключевой области на практике наличии систематического и измеримого положительного результата практически во всей организации оценивается
- 5 по 5 бальной системе
 - 10 по 10 бальной системе
 - **8 по 10 бальной системе**

32. К правилам экстремального программирования XP относятся
- Заказчик всегда рядом.
 - Весь код должен соответствовать принятому стандарту.
 - Весь код должен быть создан парным программированием.
 - Частая интеграция кода.
 - Оставлять тестирование на потом.
 - Коллективное владение кодом.
 - Оставлять оптимизацию на потом.
33. К правилам экстремального программирования XP относятся
- Любой код должен иметь Unit Test
 - Все Unit тесты должны проходить перед отдачей
 - Если найдена ошибка, то тесты корректируются или но не создаются
 - Функциональные тесты периодически выполняются и их результаты публикуются
34. Укажите основные уровни стандартизации программных средств
- Международный
 - Региональный
 - Национальный
 - Внутрифирменный
 - Внутрипрограммный
 - Внутрипроцедурный
35. Как определяется понятие «стандарт» в области программного обеспечения?
- Консенсус по спецификации, производству и использованию аппаратных и программных средств вычислительной техники
 - Соглашение по установке и применению норм и правил взаимодействия между различными программами
 - Регламент приемлемого уровня качества выпускаемого программного обеспечения
 - Множество разнообразных стандартов, процедур, методов, инструментальных средств и типов операционной среды для разработки и управления программным обеспечением
36. В чем различие между понятиями стандарта «де-факто» и «де-юре»?
- Стандарт «Де-факто» - продукт поставщика, имеющего максимальную долю на рынке
 - Стандарт «Де-юре» - продукт поставщика, имеющего минимальную долю на рынке
 - Стандарт «Де-факто» - продукт, утвержденный в качестве стандарта International Standard Organization
 - Стандарт «Де-юре» - продукт, утвержденный в качестве стандарта American National Standard Institute
37. Укажите известные вам международные организации, разрабатывающие стандарты

- ANSI
 - ISO
 - SQL
 - МЭК
 - UML
38. Почему нужны внутрифирменные стандарты?
- Потому, что они имеют узкую сферу полномочий
 - Потому, что они абсолютно конкретны
 - **Потому, что они регламентируют внутренний процесс разработки программных приложений**
 - Потому, что они базируются на лучших методиках и технологиях, используемых фирмой
39. Укажите основные стандарты, характеризующие жизненный цикл программного средства
- DOD-STD-2167A
 - **ГОСТ Р ИСО/МЭК 12207**
 - **ГОСТ ЕСПД**