

Документ подписан простой электронной подписью  
Информация о владельце:  
ФИО: Пекаревский Борис Владимирович  
Должность: Проректор по учебной и методической работе  
Дата подписания: 31.10.2023 16:51:46  
Уникальный программный ключ:  
3b89716a1076b80b2c167df0f27c09d01782ba84



МИНОБРНАУКИ РОССИИ  
федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Санкт-Петербургский государственный технологический институт  
(технический университет)»

УТВЕРЖДАЮ  
Проректор по учебной  
и методической работе  
\_\_\_\_\_ Б.В. Пекаревский  
« 24 » мая 2019 г.

**Рабочая программа дисциплины  
РАЗРАБОТКА ПРОГРАММНЫХ СИСТЕМ**

Направление подготовки

**09.03.03 Прикладная информатика**

Направленность программы бакалавриата

**Прикладная информатика в химии**

Квалификация

**Бакалавр**

Форма обучения

**Очная**

Факультет **информационных технологий и управления**

Кафедра **систем автоматизированного проектирования и управления**

Санкт-Петербург  
2019

**Б1.О.13**

## ЛИСТ СОГЛАСОВАНИЯ

Должность разработчика	Подпись	Ученое звание, фамилия, инициалы
доцент		доцент В.Н. Уланов
доцент		И.Г. Корниенко
ст. преподаватель		А.К. Федин
аспирант		Э.Э. Мусаев

Рабочая программа дисциплины «Разработка программных систем» обсуждена на заседании кафедры систем автоматизированного проектирования и управления  
протокол от «18» апреля 2019 года №9

Заведующий кафедрой

профессор Т.Б. Чистякова

Одобрено учебно-методической комиссией факультета информационных технологий и управления  
протокол от «15» мая 2019 года №9

Президент

доцент В.В. Куркина

## СОГЛАСОВАНО

Руководитель направления подготовки «Прикладная информатика»		доцент И.В. Новожилова
Директор библиотеки		Т.Н. Старостенко
Начальник методического отдела учебно-методического управления		Т.И. Богданова
Начальник учебно-методического управления		С.Н. Денисенко

## СОДЕРЖАНИЕ

1. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы.....	4
2. Место дисциплины в структуре образовательной программы.....	5
3. Объем дисциплины.....	5
4. Содержание дисциплины.....	6
4.1. Разделы дисциплины и виды занятий.....	6
4.2. Занятия лекционного типа.....	7
4.3. Занятия семинарского типа.....	11
4.3.1. Семинары, практические занятия.....	11
4.3.2. Лабораторные работы.....	12
4.4. Самостоятельная работа обучающихся.....	12
5. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине.....	16
6. Фонд оценочных средств для проведения промежуточной аттестации.....	17
7. Перечень учебных изданий, необходимых для освоения дисциплины.....	18
8. Перечень электронных образовательных ресурсов, необходимых для освоения дисциплины.....	19
9. Методические указания для обучающихся по освоению дисциплины.....	19
10. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине.....	19
10.1. Информационные технологии.....	19
10.2. Программное обеспечение.....	20
10.3. Базы данных и информационные справочные системы.....	20
11. Материально-техническое обеспечение освоения дисциплины в ходе реализации образовательной программы.....	20
12. Особенности освоения дисциплины инвалидами и лицами с ограниченными возможностями здоровья.....	20

Приложения: 1. Фонд оценочных средств для проведения промежуточной аттестации.

## 1. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы.

В результате освоения образовательной программы бакалавриата обучающийся должен овладеть следующими результатами обучения по дисциплине:

Код и наименование компетенции	Код и наименование индикатора достижения компетенции	Планируемые результаты обучения (дескрипторы)
ОПК-8 Способен принимать участие в управлении проектами создания информационных систем на стадиях жизненного цикла	ОПК-8.1 Применение основных технологий создания и внедрения информационных систем, стандартов управления жизненным циклом программной системы	Приводит примеры основных технологий создания и внедрения информационных систем, стандартов управления жизненным циклом программной системы (ЗН-1)
		Сопоставляет основные технологии создания и внедрения информационных систем, стандартов управления жизненным циклом программной системы (У-1)
		Решает задачу выбора эффективной технологии для создания и внедрения информационной системы, стандартов управления жизненным циклом программной системы (Н-1)
ПК-2 Способен разрабатывать и адаптировать прикладное программное обеспечение	ПК-2.13 Создание программной системы в соответствии с техническим заданием	Приводит примеры теоретических методов создания программной системы в соответствии с техническим заданием (ЗН-2)
		Анализирует и объясняет использование методов проектирования программной системы в соответствии с техническим заданием (У-2)
		Решает задачи создания программной системы в соответствии с техническим заданием с использованием компьютерных средств проектирования (Н-2)
	ПК-2.14 Разработка процедур интеграции программных модулей	Перечисляет принципы разработки процедур интеграции программных модулей (ЗН-3)
		Определяет закономерности при разработке процедур интеграции программных модулей (У-3)
		Демонстрирует навыки разработки процедур интеграции программных модулей проектной и рабочей технической документации по основаниям и фундаментам (Н-3)
	ПК-2.15 Проверка работоспособности программного обеспечения	Дает определения методам проверки работоспособности программного обеспечения (ЗН-4)
Разрабатывает техническую документацию проверки работоспособности программного обеспечения (У-4)		
Имеет навыки проверки работоспособности программного обеспечения (Н-4)		

## 2. Место дисциплины в структуре образовательной программы.

Дисциплина относится к дисциплинам обязательной части (Б1.О.13) и изучается на 2 курсе в 3-ом и 4-ом семестре.

В методическом плане дисциплина опирается на элементы компетенций, сформированные при изучении дисциплин «Информатика» и «Программирование». Полученные в процессе изучения дисциплины «Разработка программных систем» знания, умения и навыки могут быть использованы при изучении дисциплин «Системы тестирования программного обеспечения», «Основы разработки автоматизированных информационных систем», при прохождении производственной практики, а также при выполнении выпускной квалификационной работы.

## 3. Объем дисциплины.

Вид учебной работы	Всего, академических часов	
	Очная форма обучения	
	3 семестр	4 семестр
<b>Общая трудоемкость дисциплины</b> (зачетных единиц/ академических часов)	3/108	4/144
<b>Контактная работа с преподавателем:</b>	<b>58</b>	<b>76</b>
занятия лекционного типа	18	36
занятия семинарского типа, в т.ч.		
семинары, практические занятия	36	36
лабораторные работы		
курсовое проектирование (КР или КП)	-	-
КСР		
другие виды контактной работы		
<b>Самостоятельная работа</b>	<b>50</b>	<b>32</b>
<b>Форма текущего контроля</b> (Кр, реферат, РГР, эссе)	-	-
<b>Форма промежуточной аттестации</b> (КР, КП, зачет, экзамен)	зачет	экзамен

#### 4. Содержание дисциплины.

##### 4.1. Разделы дисциплины и виды занятий.

№ п/п	Наименование раздела дисциплины	Занятия лекционного типа, акад. часы	Занятия семинарского типа, акад. часы		Самостоятельная работа, акад. часы	Формируемые компетенции	Формируемые индикаторы
			Семинары и/или практические занятия	Лабораторные работы			
3 семестр:							
1.	Введение	2			2	ОПК-8	ОПК-8.1
2.	Алгоритмы и структуры данных	8	20		18	ПК-2	ПК-2.13
3.	Процесс проектирования	2	8		5	ПК-2	ПК-2.13
4.	Жизненный цикл программ	2			5	ОПК-8	ОПК-8.1
5.	Методология программирования	2	8		10	ПК-2	ПК-2.13
6.	Интерфейсы	2			10	ПК-2	ПК-2.14
4 семестр:							
7.	Веб-программирование	11	12		8	ПК-2	ПК-2.13
8.	Многопроцессное и многопоточное программирование	8	8		6	ПК-2	ПК-2.13
9.	Сетевое программирование	8	8		6	ПК-2	ПК-2.14
10.	Тестирование и отладка	6	8		6	ПК-2	ПК-2.15
11.	Документирование и стандартизация	3			6	ОПК-8	ОПК-8.1

#### 4.2. Занятия лекционного типа.

№ раздела дисциплины	Наименование темы и краткое содержание занятия	Объем, акад. ча- сы	Инновационная форма
1	<u>Введение</u> Основные понятия. Программно-технический комплекс. Технологии программирования. Понятие обеспечения системы. Виды обеспечения. Понятие программы, подпрограммы, сопрограммы. Свойства программы как объекта системы. Понятие составляющих программного обеспечения: программный компонент, комплекс программ, пакет программ, программный модуль, библиотека программ, программная система, сборка программ, программный продукт, программная услуга. Понятия: предметная область, конфигурация, окружение, ресурс, среда, задача, функция, функциональность, спецификация, уровень, и т.д. Классификация программ по назначению и по выполнению. тестирования: терминология тестирования, различия тестирования и отладки, фазы и технология тестирования. Критерии выбора тестов: структурные, функциональные, стохастические, мутационный, оценки покрытия проекта. Разновидности тестирования: модульное, интеграционное, системное, регрессионное, автоматизация тестирования, издержки тестирования. Тестирования инсталляции. Особенности процесса и технологии индустриального тестирования.	2	
2	<u>Алгоритмы и структуры данных</u> Алгоритмы сортировки и порядковая статистика (Пирамидальная сортировка, быстрая сортировка, сортировка за линейное время, медиана и порядковые статистики). Структуры данных: простые структуры (стек, списки, очередь, корневые деревья, хеш-таблицы, бинарные и красно-черные деревья), сложные структуры данных (B-деревья). Алгоритмы для работы с графами.	8	

№ раздела дисциплины	Наименование темы и краткое содержание занятия	Объем, акад. часы	Инновационная форма
3	<p><u>Процесс проектирования</u></p> <p>Понятие программы как изделия и основные задачи проектирования. Стадии и этапы проектирования. Перечень и содержание работ стадий и этапов. Организация процесса проектирования программного обеспечения. Методы проектирования: сверху-вниз, снизу-вверх. Схемы проектирования: каскадная, откатная, спиральная. Распределение работ между участниками проектов. Взаимодействие участников в процессе проектирования. Постановка задачи на проектирование программного обеспечения. Понятие требования к программному обеспечению. Свойства требований. Сбор, формулировка, анализ, и документирование требований к программному обеспечению. Исследование и анализ предметной области. Техническое задание (ТЗ) на разработку программного обеспечения. Состав и содержание технического задания.</p>	2	
4	<p><u>Жизненный цикл программ</u></p> <p>Понятие жизненного цикла. Модель жизненного цикла программы. Процессы жизненного цикла: основные (заказ, поставка, разработка, эксплуатация, сопровождение), вспомогательные (документирование, управление конфигурацией, обеспечение качества, верификация, аттестация, совместный анализ, аудит, решение проблем), организационные (управление проектом, создание инфраструктуры, совершенствование процессов, обучение). Состав и содержание работ каждого процесса.</p>	2	
5	<p><u>Методология программирования</u></p> <p>Понятие и виды программирования. Внутренняя структура и сегментация программ. Методологии программирования: процедурная, структурная, функциональная, логическая, объектно-ориентированная, визуальная, обобщенная. Понятие языка программирования. Классификация языков программирования. Уровни языка. Базовые составляющие языка программирования. Тенденции развития языков программирования.</p>	2	



№ раздела дисциплины	Наименование темы и краткое содержание занятия	Объем, акад. часы	Инновационная форма
6	<u>Интерфейсы</u> Понятие интерфейса. Классификация интерфейсов. Формы представления: текстовые и графические. Способы организации: командные, диалоговые, оконные, языковые. Сценарии взаимодействия: жёсткие, свободные, иерархические, прямого манипулирования. Виды диалогов: директивный, фразовый, табличный. Этапы разработки интерфейса. Эргономические требования к организации интерфейсов.	2	
7	<u>Веб-программирование</u> Особенности языка PHP и его отличия от языка C. Интеграция языков PHP и HTML. Основные типы данных. Операции. Работа со строками. Прикладные библиотеки: растровая графика и взаимодействие с СУБД MySQL. Примеры программ. Особенности языка Node.js и его отличия от языка php и C++. Интеграция языков javascript и HTML. Разработка простого Web-сервера. Особенности языка программирования Java и его отличия от языка C++. Пакеты, классы, объекты. Простые типы данных, строки, массивы. Операторы. Исключения и их обработка. События. Разработка распределенных приложений средствами CORBA. Примеры программ.	11	
8	<u>Многопроцессное и многопоточное программирование</u> Понятие процесса, его свойства и атрибуты. Цикл жизни процесса. Порождение и завершение процессов. Сигналы и реакция на них. Сигнальный механизм для синхронизации работы процессов. Обмен данных через программные каналы. Примеры программ. Обзор методов межпроцессного взаимодействия. Сообщения, семафоры и разделяемая память: создание, управление доступом, организация обмена, удаление. Отображение файлов в оперативную память. Примеры программ. Многопроцессорные ЭВМ. Структура и методика использования потоков выполнения. Создание потоков и их завершение. Объекты синхронизации потоков управления: взаимоисключающие блокировки, условные переменные, семафоры. Примеры программ.	8	

№ раздела дисциплины	Наименование темы и краткое содержание занятия	Объем, акад. часы	Инновационная форма
9	<p><u>Сетевое программирование</u>            Архитектура стека протоколов TCP/IP. Организация сетевого взаимодействия посредством механизма “гнезд” (sockets). Поточковые и дейтаграммные гнезда. Создание гнезд, привязка адреса, запросы на прием/передачу данных, завершение соединений. Примеры программ.            Механизм передачи сообщений согласно прикладному протоколу MPI (Message-Passing Interface). Взаимодействие “один к одному” и коллективное взаимодействие. Синхронные и асинхронные операции. Представление в сообщениях разнотипных данных. Группирование процессов. Виртуальные топологии процессов. Средства выполнения и мониторинга MPI-программ. Примеры программ.</p>	8	
10	<p><u>Тестирование и отладка</u>            Понятие ошибки. Основные виды ошибок: синтаксические, алгоритмические, структурные, концептуальные. Понятия тестирования. Основные понятия: случай, контрольные данные, тест-план, протокол, покрытие. Концепции чёрного, серого, белого ящика при тестировании. Методы тестирования: индукции, дедукции, ручной, обратного прослеживания. Составление тестовых планов. Поиск ошибок в программном коде. Отладка.</p>	6	
11	<p><u>Документирование и стандартизация</u>            Документирование программного обеспечения. ЕСПД и её содержание. Основы составления руководства пользователя и руководства администратора. Разработка справочных систем в формате HTML/CHM. Оформление и комментирование программного кода. Подготовка презентаций на программное обеспечение.</p>	3	

### 4.3. Занятия семинарского типа.

#### 4.3.1. Семинары, практические занятия.

№ раздела дис- циплины	Наименование темы и краткое содержание занятия	Объем, акад. часы	Инновационная форма
		всего	
2	<u>Алгоритмы и структуры данных</u> Разбор программной реализации алгоритмов сортировки (сортировка выбором, сортировка пузырьком, сортировка вставками, сортировка слиянием, сортировка Шелла, быстрая сортировка). Разбор программной реализации структур данных (стеки, очереди, связанные списки, корневые деревья, бинарные деревья, В-деревья).	12	Слайд-презентация, групповая дискуссия
3	<u>Процесс проектирования</u> Разбор примеров постановки задач на проектирование программного обеспечения. Анализ технического задания на разработку программного обеспечения, его состав и содержание.	8	Слайд-презентация
5	<u>Методология программирования</u> Ознакомление с методологиями программирования (процедурная, структурная, функциональная, логическая, объектно-ориентированная, визуальная, обобщенная).	8	Слайд-презентация
7	<u>Веб-программирование</u> Разбор примеров программ на PHP. Разбор примера простого web-сервера на Node.js. Разбор примеров программ на Java.	12	Слайд-презентация, групповая дискуссия
8	<u>Многопроцессное и многопоточное программирование</u> Разбор примеров многопроцессных и многопоточных программ.	8	Слайд-презентация, групповая дискуссия
9	<u>Сетевое программирование</u> Разбор примеров программ с сетевым взаимодействием посредством sockets. Разбор примеров MPI-программ.	8	Слайд-презентация, групповая дискуссия
10	<u>Тестирование и отладка</u> Ознакомление с методами тестирования. Разбор примеров поиска ошибок в программном коде и отладке.	8	Слайд-презентация

### 4.3.2. Лабораторные работы.

Не предусмотрены.

### 4.4. Самостоятельная работа обучающихся.

№ раздела дисциплины	Перечень вопросов для самостоятельного изучения	Объем, акад. часы	Форма контроля
1	<u>Введение</u> Терминология, используемая в разработке программных систем. Классификация программ по назначению и по выполнению.	2	Выступление на занятиях с докладом
2	<u>Алгоритмы и структуры данных</u> Хеширование и хеш-таблицы. Бинарные деревья. Красно-черные деревья. Динамические порядковые статистики. Деревья отрезков. В-деревья.	18	Тестирование №1
3	<u>Процесс проектирования</u> Методы проектирования программных систем. Распределение работ между участниками проектов. Требования, предъявляемые к программному обеспечению. Требования в SWEBOOK. Требования в RUP. Схема Лефингвелла. IEEE 830. ГОСТ 34 серии.	5	Выступление на занятиях с докладом
4	<u>Жизненный цикл программ</u> ГОСТ 34.601-90. ISO/IEC 12207:2008. Модель жизненного цикла программного обеспечения. Процессы жизненного цикла программного обеспечения. Стадии жизненного цикла программного обеспечения.	5	Выступление на занятиях с докладом
5	<u>Методология программирования</u> Методология императивного программирования. Методология ООП. Методология функционального программирования. Методология логическое программирование. Методология программирования в ограничениях. Классификация языков программирования.	10	Тестирование №2
6	<u>Интерфейсы</u> Виды интерфейсов пользователя. Эргономические требования к организации интерфейсов. Текстовые редакторы. Типовая структура интерфейса. Графические процессоры. Электронные таблицы.	10	Выступление на занятиях с докладом
7	<u>Веб-программирование</u> Синтаксис языка PHP. Переменные. Управляющие структуры. Массивы. Функции пользователя. Объектная модель. Особенности языка Node.js. Свойства языка Java. Отличия Java от C. Классы и объекты в Java.	8	Выступление на занятиях с докладом

№ раздела дисциплины	Перечень вопросов для самостоятельного изучения	Объем, акад. часы	Форма контроля
8	<u>Многопроцессное и многопоточное программирование</u> Программирование для систем с разделяемой памятью. Программирование для систем с передачей сообщений. Многопоточность на C++. Семафоры. Приоритеты. Очереди, FIFO, LIFO и многопоточность.	6	Тестирование №3
9	<u>Сетевое программирование</u> Сетевые серверы. Сетевые клиенты. Архитектура стека протоколов TCP/IP. Организация сетевого взаимодействия посредством механизма "гнезд" (sockets). Протокол MPI.	6	Выступление на занятиях с докладом
10	<u>Тестирование и отладка</u> Стандартный глоссарий терминов, используемых в тестировании программного обеспечения. Классификация видов тестирования. Функциональные виды тестирования. Нефункциональные виды тестирования. Связанные с изменениями виды тестирования. Уровни тестирования программного обеспечения.	6	Выступление на занятиях с докладом
11	<u>Документирование и стандартизация</u> Документация, используемая в разработке программного обеспечения. Документ руководство пользователя и руководства администратора. Средства разработка справочных систем. Оформление и комментирование программного кода.	6	Выступление на занятиях с докладом

#### **4.4.1. Темы докладов.**

В качестве тем для докладов по 1 разделу «Введение», могут быть рекомендованы следующие темы:

1. Терминология, используемая в разработке программных систем.
2. Классификация программ по назначению и по выполнению.
3. Российский IT-сектор.

В качестве тем для докладов по 3 разделу «Процесс проектирования», могут быть рекомендованы следующие темы:

1. Методы проектирования программных систем.
2. Требования в SWEBOOK.
3. Требования в RUP.
4. Схема Лефингвелла.
5. IEEE 830.
6. ГОСТ 34 серии.

В качестве тем для докладов по 4 разделу «Жизненный цикл программ», могут быть рекомендованы следующие темы:

1. Каскадная модель жизненного цикла программного обеспечения.
2. Разработка через тестирование.
3. Спиральная модель жизненного цикла программного обеспечения.
4. Экстремальное программирование (XP), SCRUM, инкрементальная модель.

В качестве тем для докладов по 6 разделу «Интерфейсы», могут быть рекомендованы следующие темы:

1. Эргономические требования к организации интерфейсов.
2. Текстовые редакторы.
3. Графические процессоры.
4. Электронные таблицы.
5. Издательские системы.

В качестве тем для докладов по 7 разделу «Веб-программирование», могут быть рекомендованы следующие темы:

1. Графика в PHP.
2. PHP и СУБД MySQL.
3. Отличия Java от C.
4. Особенности языка Node.js.
5. Интеграция языков javascript и HTML.

В качестве тем для докладов по 9 разделу «Сетевое программирование», могут быть рекомендованы следующие темы:

1. Сетевые серверы.
2. Сетевые клиенты.
3. Архитектура стека протоколов TCP/IP.
4. Организация сетевого взаимодействия посредством механизма “гнезд” (sockets).
5. Протокол MPI.

В качестве тем для докладов по 10 разделу «Тестирование и отладка», могут быть рекомендованы следующие темы:

1. Стандартный глоссарий терминов, используемых в тестировании программного обеспечения.
2. Классификация видов тестирования.
3. Уровни тестирования программного обеспечения.
4. Средства автоматизированного тестирования.

В качестве тем для докладов по 11 разделу «Документирование и стандартизация», могут быть рекомендованы следующие темы:

1. Документация, используемая в разработке программного обеспечения.
2. Документ руководство пользователя и руководства администратора.
3. Средства разработка справочных систем.
4. Оформление и комментирование программного кода.

#### 4.4.2 Тестирование

В качестве примера оценочных средств для текущего контроля успеваемости студентов по 2 разделу «Алгоритмы и структуры данных» представлены следующие варианты заданий:

1) Открытые тесты (необходимо вписать слово в предложение):

- Структура \_\_\_\_\_ - это способ хранения и организации данных, облегчающий доступ к этим данным и их модификацию.

- \_\_\_\_\_ - элементарная структура данных реализующая стратегию «последним вошел – первым вышел».

- Связанный \_\_\_\_\_ - это структура данных, в которой объекты расположены в линейном порядке, где порядок определяется указателями на каждый объект.

2) Закрытые тесты (необходимо выбрать один или несколько вариантов ответа):

- Необходимо выбрать правильный ответ. Структура данных реализующая стратегию «первый вошел – первый вышел» - это

· стек;

· очередь.

- Необходимо выбрать правильный ответ. Способ обхода бинарного дерева поиска, при котором сначала выводится корень, а затем значения правого и левого поддеревьев:

· центрированный обход дерева;

· обход в прямом порядке;

· обход в обратном порядке.

- Необходимо выбрать правильный ответ. Красно черные деревья:

· сбалансированные;

· приближенно сбалансированные (путь от корня не отличается от другого пути более чем в два раза);

· несбалансированный.

В качестве примера оценочных средств для текущего контроля успеваемости студентов по 5 разделу «Методология программирования» представлены следующие варианты заданий:

1) Открытые тесты (необходимо вписать слово в предложение):

- \_\_\_\_\_ программирование — парадигма программирования, в которой процесс вычисления трактуется как вычисление значений функций в математическом понимании последних.

- \_\_\_\_\_ программирование — это парадигма программирования, которая описывает процесс вычисления в виде инструкций, изменяющих состояние данных.

- \_\_\_\_\_ — свойство системы, позволяющее объединить данные и методы, работающие с ними, в классе.

2) Закрытые тесты (необходимо выбрать один или несколько вариантов ответа):

- Необходимо выбрать правильный ответ. Какой из языков программирования относится к логическому программированию:

· C++;

· C#;

· Prolog;

· Java.

- Необходимо выбрать правильный ответ. Какой из языков программирования относится к логическому программированию:

· C++;

· C#;

· Prolog;

· Java;

· JavaScript.

В качестве примера оценочных средств для текущего контроля успеваемости студентов по 8 разделу «Многопроцессное и многопоточное программирование» представлены следующие варианты заданий:

1) Открытые тесты (необходимо вписать слово в предложение):

- \_\_\_\_\_ — свойство платформы или приложения, состоящее в том, что процесс, порождённый в операционной системе, может состоять из нескольких потоков, выполняющихся «параллельно», то есть без предписанного порядка во времени.

- \_\_\_\_\_ — это API-интерфейс, который является отраслевым стандартом для создания параллельных приложений для компьютеров с совместным использованием памяти..

- \_\_\_\_\_ позволяет задавать критические секции кода, в которые в каждый отдельный момент может входить только один поток, гарантируя, что любые временные недействительные состояния вашего объекта будут невидимы его клиентам..

2) Закрытые тесты (необходимо выбрать один или несколько вариантов ответа):

- Необходимо выбрать один или несколько правильных ответов. Когда использовать потоки:

- приложениям требуется выполнять несколько задач одновременно;
- упрощения структуры сложных систем (использование очередей и асинхронной обработки);
- наличие задач подходящих для фоновой обработки.

## **5. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине.**

Методические указания для обучающихся по организации самостоятельной работы по дисциплине, включая перечень тем самостоятельной работы, формы текущего контроля по дисциплине и требования к их выполнению размещены в электронной информационно-образовательной среде СПбГТИ(ТУ) на сайте: <http://media.technolog.edu.ru>



## **6. Фонд оценочных средств для проведения промежуточной аттестации.**

Своевременное выполнение обучающимся мероприятий текущего контроля позволяет превысить (достигнуть) пороговый уровень («удовлетворительно») освоения предусмотренных элементов компетенций.

Результаты дисциплины считаются достигнутыми, если для всех элементов компетенций превышен (достигнут) пороговый уровень освоения компетенции на данном этапе.

Промежуточная аттестация по дисциплине проводится в форме зачета.

К сдаче зачета допускаются студенты, выполнившие все формы текущего контроля.

Зачет предусматривают выборочную проверку освоения предусмотренных элементов компетенций и комплектуются вопросами (заданиями) двух видов: один теоретический вопрос (для проверки знаний) и комплексная задача (для проверки умений и навыков).

При сдаче зачета, студент получает два вопроса из перечня вопросов, время подготовки студента к устному ответу - до 30 мин.

Пример варианта вопросов на зачете:

### **Билет №1**

1. Постановка задачи на проектирование программного обеспечения.
2. Алгоритм сортировки выбором. Пример программной реализации.

Заключительная аттестация по дисциплине проводится в форме экзамена.

Экзамен предусматривают выборочную проверку освоения предусмотренных элементов компетенций и комплектуются вопросами (заданиями) двух видов: теоретический вопрос (для проверки знаний) и комплексная задача (для проверки умений и навыков).

При сдаче экзамена, студент получает три вопроса из перечня вопросов, время подготовки студента к устному ответу - до 45 мин.

Пример варианта вопросов на экзамене:

### **Вариант № 1**

1. Многопоточность на C++.
2. Концепции чёрного, серого, белого ящика при тестировании
3. Алгоритм сортировки выбором. Пример программной реализации

Фонд оценочных средств по дисциплине представлен в Приложении № 1

Результаты освоения дисциплины считаются достигнутыми, если для всех элементов компетенций достигнут пороговый уровень освоения компетенции на данном этапе – оценка «удовлетворительно».

Фонд оценочных средств по дисциплине представлен в Приложении № 1

## **7. Перечень учебных изданий, необходимых для освоения дисциплины.**

### **а) печатные издания:**

1. Норенков, И. П. Автоматизированные информационные системы: учеб. пособие для вузов / И. П. Норенков. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2011. – 342 с.
2. Чистякова, Т. Б. Применение универсальных моделирующих программ для синтеза и анализа технологических процессов: учеб. пособие / Т. Б. Чистякова, Л. В. Гольцева, А. В. Козлов. – СПб. : СПбГТИ(ТУ), 2011. – 66 с.
3. Орлов, С. А. Теория и практика языков программирования : учебник по направлению "Информатика и вычислительная техника" / С. А. Орлов. – М. ; СПб. ; Н. Новгород : Питер, 2014. - 688 с.
4. Бройдо, В. Л. Вычислительные системы, сети и телекоммуникации : учеб. пособие для вузов / В. Л. Бройдо, О. П. Ильина. – 4-е изд. – М. ; СПб. ; Н. Новгород : Питер, 2011. – 554 с.
5. Ликнесс, Д. Приложения для Windows 8 на C# и XAML / Дж. Ликнесс. - М. ; СПб. ; Н. Новгород : Питер, 2013. - 368 с.
6. Советов, Б. Я. Представление знаний в информационных системах : учеб. для вузов / Б. Я. Советов, В. В. Цехановский, В. Д. Чертовский. – М. : Академия, 2011. – 143 с.
7. Макконнелл, С. Совершенный код. Мастер-класс / С. Макконнелл ; пер. с англ. под ред. В. Г. Вшивцева. – М. : Рус. ред., 2012. - 867 с.
8. Петкович, Д. Microsoft SQL Server 2008. Руководство для начинающих : пер. с англ. / Д. Петкович. – СПб. : БХВ-Петербург, 2012. - 730 с.
9. Дейтел, П. Как программировать на Visual C# 2012 : Включая работу в Windows 7 и Windows 8 / П. Дейтел, Х. Дейтел. - 5-е изд. - М. ; СПб. ; Н. Новгород : Питер, 2014. - 864 с.
10. Лафоре, Р. Объектно-ориентированное программирование в C++ / Р. Лафоре. - 4-е изд. - М. ; СПб. ; Н. Новгород : Питер, 2015. - 928 с.
11. Лав, Р. Linux. Системное программирование / Р. Лав. - 2-е изд. - М. ; СПб. ; Н. Новгород : Питер, 2014. - 448 с.
12. Павловская, Т. А. C#. Программирование на языке высокого уровня : учеб. пособие для вузов / Т. А. Павловская. - М. ; СПб. ; Н. Новгород : Питер, 2014. - 432 с.
13. Чистякова, Т. Б. Программирование на языках высокого уровня. Базовый курс : учеб. пособие / Т. Б. Чистякова, Р. В. Антипин, И. В. Новожилова ; СПбГТИ(ТУ). Каф. систем автоматизир. проектирования и упр. – СПб. : [б. и.], 2008. – 101 с.
14. Тенишев, Д. Ш. Лингвистическое и программное обеспечение автоматизированных систем: учеб. пособие для вузов / Д. Ш. Тенишев ; под ред. Т. Б. Чистяковой. – СПб. : Центр образовательных программ «Профессия», 2010. – 403 с.
15. Головин, Ю. А. Информационные сети: учеб. для вузов / Ю. А. Головин, А. А. Суконщиков, С. А. Яковлев. – М.: Академия, 2011. – 376 с.
16. Елович, И. В. Информатика : Учебник для вузов по техническим и естественнонаучным направлениям / И. В. Елович, И. В. Кулибаба; под ред. Г. Г. Раннева. - М. : Академия, 2011. - 394 с.
17. Уильямс, Б. WordPress для профессионалов. Разработка и дизайн сайтов / Б. Уильямс, Д. Дэмстра, Х. Стэрн. - М. ; СПб. ; Н. Новгород : Питер, 2014. - 464 с. : ил. - (Для профессионалов).

### **б) электронные учебные издания:**

18. Царев, Р. Ю. Проектирование, разработка и оценка надежности сложных программных систем : монография / Р. Ю. Царев. – Красноярск : КрасГАУ, 2017. – 232 с. (ЭБС «Лань»)
19. Барков, И. А. Объектно-ориентированное программирование : учебник / И. А. Барков. – Санкт-Петербург : Лань, 2019. – 700 с. (ЭБС «Лань»)

## **8. Перечень электронных образовательных ресурсов, необходимых для освоения дисциплины.**

учебный план, РПД и учебно-методические материалы: <http://media.technolog.edu.ru>  
веб-страница журнала «Информационные технологии» <http://www.novtex.ru/IT>  
сайты информационных технологий: <http://inftech.webservis.ru>, <http://citforum.ru>  
информационно-аналитический портал «Научная электронная библиотека»  
<http://elibrary.ru>

международные мультидисциплинарные аналитические реферативные базы данных научных публикаций <http://webofknowledge.com>, <http://scopus.com>

электронно-библиотечные системы:

«Электронный читальный зал – БиблиоТех» <https://technolog.bibliotech.ru/>;

«Лань» <https://e.lanbook.com/books/>.

## **9. Методические указания для обучающихся по освоению дисциплины.**

Все виды занятий по дисциплине «Информационные системы управления качеством в автоматизированных и автоматических производствах» проводятся в соответствии с требованиями следующих СТП:

СТП СПбГТИ 040-02. КС УКДВ. Виды учебных занятий. Лекция. Общие требования;

СТО СПбГТИ 018-2014. КС УКДВ. Виды учебных занятий. Семинары и практические занятия. Общие требования к организации и проведению.

СТП СПбГТИ 048-2009. КС УКДВ. Виды учебных занятий. Самостоятельная планируемая работа студентов. Общие требования к организации и проведению.

Планирование времени, необходимого на изучение данной дисциплины, лучше всего осуществлять на весь семестр, предусматривая при этом регулярное повторение пройденного материала.

Основными условиями правильной организации учебного процесса для студентов является:

плановость в организации учебной работы;

серьезное отношение к изучению материала;

постоянный самоконтроль.

На занятия студент должен приходить, имея багаж знаний и вопросов по уже изученному материалу.

## **10. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине.**

### **10.1. Информационные технологии.**

В учебном процессе по данной дисциплине предусмотрено использование информационных технологий:

чтение лекций с использованием слайд-презентаций;

взаимодействие с обучающимися посредством ЭИОС.

## 10.2. Программное обеспечение.

Операционная система Microsoft Windows 10  
Интегрированная среда разработки программного обеспечения: Microsoft Visual Studio 2019, Android Studio 4.2  
Система управления базами данных Microsoft Access 2018  
Редактор диаграмм и блок-схем Microsoft Visio 2019  
Офисный пакет программ LibreOffice

## 10.3. Базы данных и информационные справочные системы.

Справочно-поисковая система «Консультант-Плюс»

## 11. Материально-техническое обеспечение освоения дисциплины в ходе реализации образовательной программы.

На кафедре систем автоматизированного проектирования и управления СПбГТИ(ТУ) имеется необходимая материально-техническая база, соответствующая действующим санитарным и противопожарным правилам и нормам:

Наименование компьютерного класса кафедры	Оборудование
Класс информационных и интеллектуальных систем	Персональные компьютеры (20 шт.): четырехядерный процессор Intel Core i7-920 (2666 МГц), ОЗУ 6 Гб; НЖМД 250 Гб; CD/DVD привод, DVD-RW; видеокарта NVIDIA GeForce GT 220 (1024 Мб); звуковая и сетевая карты, встроенные в материнскую плату.
Класс интегрированных систем проектирования и управления химико-технологическими процессами	Персональные компьютеры (15 шт.): двухядерный процессор Intel Core 2 Duo (2,33 ГГц); ОЗУ 4096 Мб; НЖМД 250 Гб; CD/DVD привод, DVD-RW; видеокарта NVIDIA GeForce 8500 GT; звуковая и сетевая карты, встроенные в материнскую плату.
Класс базовых информационных процессов и технологий	Персональные компьютеры (9 шт.): моноблок Lenovo C360 с 19,5-дюймовым дисплеем; процессор Intel Core i3-4130T (2,9ГГц); ОЗУ 4 Гб; НЖМД 1000 Гб; встроенные DVD-RW, видеокарта Intel HD Graphics 4400, звуковая и сетевая карты.
Лекционная аудитория	Учебная мебель. Мультимедийный проектор NEC NP41. Ноутбук Asus абј на базе процессора Intel Core Duo T2000. Мультимедийная интерактивная доска ScreenMedia.

## 12. Особенности освоения дисциплины инвалидами и лицами с ограниченными возможностями здоровья.

Для инвалидов и лиц с ограниченными возможностями учебные процесс осуществляется в соответствии с Положением об организации учебного процесса для обучения инвалидов и лиц с ограниченными возможностями здоровья СПбГТИ(ТУ), утвержденным ректором 28.08.2014.

**Фонд оценочных средств  
для проведения промежуточной аттестации по  
дисциплине «РАЗРАБОТКА ПРОГРАММНЫХ СИСТЕМ»**

**1. Перечень компетенций и этапов их формирования.**

Индекс компетенции	Содержание	Этап формирования
ОПК-8	Способен принимать участие в управлении проектами создания информационных систем на стадиях жизненного цикла	начальный
ПК-2	Способен разрабатывать и адаптировать прикладное программное обеспечение	промежуточный

## 2. Показатели и критерии оценивания компетенций на различных этапах их формирования, шкала оценивания

Код и наименование индикатора достижения компетенции	Показатели сформированности (дескрипторы)	Критерий оценивания	Уровни сформированности (описание выраженности дескрипторов)		
			«удовлетворительно» (пороговый)	«хорошо» (средний)	«отлично» (высокий)
ОПК-8.1 Применение основных технологий создания и внедрения программных систем, стандартов управления жизненным циклом программной системы	Приводит примеры основных технологий создания и внедрения программных систем и стандартов управления жизненным циклом программной системы (ЗН-1)	Правильные ответы на вопросы №1-9 к зачету	Допускает ошибки при перечислении основных технологий создания и внедрения программных систем, стандартов управления жизненным циклом программной системы.	Перечисляет основные технологии создания и внедрения программных систем без ошибок, но путается в последовательности проектирования.	Идеально ориентируется в основных технологиях создания и внедрения программных систем и последовательности их проектирования в соответствии со стандартами управления жизненным циклом.
	Сопоставляет основные технологии создания и внедрения программных систем и стандарты управления жизненным циклом программной системы (У-1)	Правильные ответы на вопросы №1-9 к зачету	Путается при сопоставлении основных технологий создания и внедрения программных систем. Делает ошибки в использовании стандартов управления жизненным циклом программной системы	Составляет и делает выводы по выбору технологий создания и внедрения программных систем. Допускает неточности в использовании стандартов управления жизненным циклом программной системы	Сопоставляет основные технологии создания и внедрения программных систем и стандарты управления жизненным циклом программной системы
	Решает задачу выбора эффективной технологии для создания и внедрения программной системы и стандартов управления жизненным циклом программной системы (Н-1)	Правильные ответы на вопросы №1-9	Допускает ошибки при решении задачи выбора эффективной технологии для создания и внедрения программной системы и стандартов управления жизненным циклом.	Допускает неточности при решении задачи выбора эффективной технологии для создания и внедрения информационной системы и стандартов управления жизненным циклом.	Идеально решает задачу выбора эффективной технологии для создания и внедрения программной системы в соответствии со стандартами управления жизненным циклом.
ПК-2.13 Создание программной системы в соответствии с техническим заданием	Приводит примеры теоретических методов создания программной системы в соответствии с техническим заданием (ЗН-2)	Правильные ответы на вопросы №10-23 к зачету	Допускает ошибки при перечислении теоретических методов создания программной системы в соответствии с техническим заданием	В целом правильно приводит примеры методов создания программной системы в соответствии с техническим заданием с небольшими подсказками преподавателя	Уверенно и безошибочно приводит примеры методов создания программной системы в соответствии с техническим заданием
	Анализирует и объясняет использование методов проектирования программной системы в соответствии с техническим заданием.(У-2)	Правильные ответы на вопросы №10-23 к зачету	Путается при анализе и объяснении использования методов проектирования программной системы в соответствии с техническим заданием.	В основном правильно проводит анализ методов проектирования программной системы в соответствии с техническим заданием, но допускает неточности в объяснении их использования	Легко ориентируется в использовании методов проектирования программной системы в соответствии с техническим заданием, проводит полный анализ и объясняет применимость.
	Решает задачи создания программной системы в соответствии с техническим заданием с использованием компьютерных средств проектирования.(Н-2)	Правильные ответы на вопросы №10-23 к зачету	Слабо ориентируется в алгоритме решения задачи создания программной системы в соответствии с техническим заданием с использованием компьютерных средств проектирования.	Выполняет алгоритм решения задачи создания программной системы в соответствии с техническим заданием с использованием компьютерных средств проектирования с небольшими ошибками	Быстро, качественно и без ошибок решает задачу создания программной системы в соответствии с техническим заданием с использованием компьютерных средств проектирования
ПК-2.14 Разработка процедур	Перечисляет принципы разработки процедур интеграции программных модулей (ЗН-3)	Правильные ответы на вопросы № 24-44 к зачету	Путается в перечислении принципов разработки процедур интеграции программных модулей.	Перечисляет принципы разработки процедур интеграции программных модулей с небольшими ошибками.	Уверенно и без ошибок перечисляет принципы разработки процедур интеграции программных модулей.

Код и наименование индикатора достижения компетенции	Показатели сформированности (дескрипторы)	Критерий оценивания	Уровни сформированности (описание выраженности дескрипторов)		
			«удовлетворительно» (пороговый)	«хорошо» (средний)	«отлично» (высокий)
интеграции программных модулей	Определяет закономерности при разработке процедур интеграции программных модулей (У-3)	Правильные ответы на вопросы № 24-44 к зачету	Неточно и с ошибками перечисляет закономерности при разработке процедур интеграции программных модулей.	Определяет закономерности при разработке процедур интеграции программных модулей с небольшими подсказками преподавателя.	Идеально ориентируется в закономерностях при разработке процедур интеграции программных модулей.
	Демонстрирует навыки разработки процедур интеграции программных модулей.(Н-3)	Правильные ответы на вопросы № 24-44 к зачету	Путается в разработке процедур интеграции программных модулей.	Демонстрирует с незначительными ошибками навыки разработки процедур интеграции программных модулей.	Уверенно демонстрирует прекрасные навыки разработки процедур интеграции программных модулей.
ПК-2.15 Проверка работоспособности программного обеспечения	Дает определения методам проверки работоспособности программного обеспечения (ЗН-4)	Правильные ответы на вопросы №45-74 к зачету	Допускает ошибки при определении методов проверки работоспособности программного обеспечения.	В целом правильно приводит примеры методов проверки работоспособности программного обеспечения, допуская небольшие недочеты.	Абсолютно правильно и с пояснениями приводит примеры методов тестирования при проверке работоспособности программного обеспечения.
	Разрабатывает техническую документацию проверки работоспособности программного обеспечения (У-4)	Правильные ответы на вопросы № 45-74 к зачету	Имеет представление о разработке технической документации проверки работоспособности программного обеспечения.	Разрабатывает техническую документацию проверки работоспособности программного обеспечения с небольшими ошибками.	Разрабатывает рабочую техническую документацию проверки работоспособности программного обеспечения, может оформлять законченные проектно-конструкторские работы.
	Имеет навыки проверки работоспособности программного обеспечения (Н-4)	Правильные ответы на вопросы № 45-74 к зачету	Имеет слабые навыки разработки проектной и рабочей технической документации проверки работоспособности программного обеспечения	Имеет навыки разработки проектной и рабочей технической документации проверки работоспособности программного обеспечения, но допускает незначительные ошибки	Демонстрирует уверенные навыки разработки проектной и рабочей технической документации тестирования проверки работоспособности программного обеспечения.

### **3. Типовые контрольные задания для проведения промежуточной аттестации**

#### **а) Вопросы для оценки сформированности элементов компетенции ОПК-8:**

1. Основные документы программного обеспечения.
2. Документы программного обеспечения: ГОСТ 34 серии.
3. Содержание руководства пользователя и руководства администратора.
4. Постановка задачи на проектирование программного обеспечения.
5. Понятие требования к программному обеспечению. Свойства требований.
6. Сбор, формулировка, анализ и документирование требований к программному обеспечению.
7. Понятие жизненного цикла. Модель жизненного цикла программы.
8. Процессы жизненного цикла: основные, вспомогательные, организационные.

Состав и содержание работ каждого процесса.

9. Перечень и содержание работ стадий и этапов.

#### **б) Вопросы для оценки сформированности элементов компетенции ПК-2:**

10. Алгоритм сортировки выбором. Пример программной реализации.
11. Алгоритм быстрой сортировки. Пример программной реализации.
12. Алгоритм сортировки пузырьком. Пример программной реализации.
13. Алгоритм сортировки вставками. Пример программной реализации.
14. Алгоритм сортировки слиянием. Пример программной реализации.
15. Алгоритмы сортировки Шелла. Пример программной реализации.
16. Алгоритмы поиска кратчайшего пути в графе
17. Алгоритм волновой трассировки Ли.
18. Алгоритм Дейкстры.
19. Алгоритм A\*.
20. Алгоритм Беллмана-Форда.
21. Алгоритм Джонсона.
22. Алгоритм Флойда-Уоршелла.
23. Алгоритм Левита.
24. Исследование и анализ предметной области.
25. Эргономические требования к организации интерфейсов.
26. Основные понятия: Программно-технический комплекс. Технологии программирования. Понятие обеспечения системы. Виды обеспечения.
27. Основные понятия: Понятие программы, подпрограммы, сопрограммы. Свойства программы как объекта системы.
28. Понятие составляющих программного обеспечения: программный компонент, комплекс программ, пакет программ, программный модуль, библиотека программ, программная система, сборка программ, программный продукт, программная услуга.
29. Понятия: предметная область, конфигурация, окружение, ресурс, среда, задача, функция, функциональность, спецификация, уровень.
30. Классификация программ по назначению и по выполнению.
31. Понятие программы как изделия и основные задачи проектирования. Стадии и этапы проектирования.
32. Организация процесса проектирования программного обеспечения.
33. Методы проектирования: сверху-вниз, снизу-вверх. Схемы проектирования: каскадная, откатная, спиральная.
34. Распределение работ между участниками проектов. Взаимодействие участников в процессе проектирования.
35. Понятие и виды программирования. Внутренняя структура и сегментация программ.
36. Методологии программирования: процедурная, структурная, функциональная, логическая, объектно-ориентированная, визуальная, обобщенная.
37. Понятие языка программирования. Классификация языков программирования.



38. Уровни языка. Базовые составляющие языка программирования.
39. Понятие интерфейса. Классификация интерфейсов. Формы представления: текстовые и графические.
40. Способы организации: командные, диалоговые, оконные, языковые. Сценарии взаимодействия: жёсткие, свободные, иерархические, прямого манипулирования.
41. Виды диалогов: директивный, фразовый, табличный. Этапы разработки интерфейса.
42. Особенности языка PHP и его отличия от языка C. Интеграция языков PHP и HTML. Основные типы данных. Операции. Работа со строками.
43. Прикладные библиотеки: растровая графика и взаимодействие с СУБД MySQL. Примеры программ на PHP.
44. Особенности языка Node.js и его отличия от языка php и C++. Интеграция языков javascript и HTML. Разработка простого Web-сервера.
45. Особенности языка программирования Java и его отличия от языка C++. Пакеты, классы, объекты.
46. Особенности языка программирования Java. Простые типы данных, строки, массивы. Операторы. Исключения и их обработка. События.
47. Разработка распределенных приложений JAVA средствами CORBA. Примеры программ.
48. Понятие процесса, его свойства и атрибуты. Цикл жизни процесса. Порождение и завершение процессов.
49. Сигналы и реакция на них. Сигнальный механизм для синхронизации работы процессов. Обмен данных через программные каналы. Примеры программ.
50. Методы межпроцессного взаимодействия. Сообщения, семафоры и разделяемая память: создание, управление доступом, организация обмена, удаление. Отображение файлов в оперативную память. Примеры программ.
51. Многопроцессорные ЭВМ. Структура и методика использования потоков выполнения. Создание потоков и их завершение.
52. Объекты синхронизации потоков управления: взаимоисключающие блокировки, условные переменные, семафоры. Примеры программ.
53. Архитектура стека протоколов TCP/IP.
54. Организация сетевого взаимодействия посредством механизма “гнезд” (sockets). Поточковые и дейтаграммные гнезда.
55. Создание гнезд, привязка адреса, запросы на прием/передачу данных, завершение соединений. Примеры программ.
56. Механизм передачи сообщений согласно прикладному протоколу MPI (Message-Passing Interface). Взаимодействие “один к одному” и коллективное взаимодействие. Примеры программ.
57. Синхронные и асинхронные операции. Представление в сообщениях разнотипных данных. Примеры программ.
58. Группирование процессов. Виртуальные топологии процессов. Средства выполнения и мониторинга MPI-программ. Примеры программ.
59. Состав и содержание технического задания на разработку программного обеспечения.
60. Методы разработки справочных систем.
61. Оформление и комментирование программного кода. Примеры стилей оформления.
62. Структура презентации на программное обеспечение.
63. реализации.
64. Структура данных стек. Пример программной реализации.
65. Структура данных очередь. Пример программной реализации.
66. Структура данных список. Пример программной реализации.

67. Структура данных дерево поиска. Пример программной реализации.
68. Сложные структуры данных. В-деревья.
69. Понятие ошибки. Основные виды ошибок: синтаксические, алгоритмические, структурные, концептуальные.
70. Понятия тестирования. Основные понятия: случай, контрольные данные, тест-план, протокол, покрытие.
71. Концепции чёрного, серого, белого ящика при тестировании.
72. Методы тестирования: индукции, дедукции, ручной, обратного прослеживания.
73. Составление тестовых планов.
74. Поиск ошибок в программном коде. Отладка.

**5. Методические материалы для определения процедур оценивания знаний, умений и навыков, характеризующих этапы формирования компетенций.**

Промежуточная аттестация по дисциплине проводится в соответствии с требованиями СПб ГТИ(ТУ) 016-2015. КС УКДВ Порядок проведения зачетов и экзаменов.

По дисциплине промежуточная аттестация проводится в форме защиты курсового проекта (курсовой работы), экзамена или зачёта.

Шкала оценивания на экзамене балльная («отлично», «хорошо», «удовлетворительно», «неудовлетворительно»), на зачёте – «зачёт», «незачет». При этом «зачёт» соотносится с пороговым уровнем сформированности компетенции.