МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ»
(ТУСУР)

УТВЕРЖДАЮ
Директор департамента образования
_____П. Е. Троян
«___»_____20__г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Языки программирования

Уровень образования: **высшее образование - специалитет**
Направление подготовки / специальность: **10.05.02 Информационная безопасность телекоммуникационных систем**
Направленность (профиль) / специализация: **Защита информации в системах связи и управления**
Форма обучения: **очная**
Факультет: **ФБ, Факультет безопасности**
Кафедра: **БИС, Кафедра безопасности информационных систем**
Курс: **2**
Семестр: **3**
Учебный план набора 2016 года

Распределение рабочего времени

№	Виды учебной деятельности	3 семестр	Всего	Единицы
1	Лекции	36	36	часов
2	Практические занятия	36	36	часов
3	Всего аудиторных занятий	72	72	часов
4	Из них в интерактивной форме	20	20	часов
5	Самостоятельная работа	36	36	часов
6	Всего (без экзамена)	108	108	часов
7	Подготовка и сдача экзамена	36	36	часов
8	Общая трудоемкость	144	144	часов
		4.0	4.0	З.Е.

Экзамен: 3 семестр

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Шелупанов А.А.
Должность: Ректор
Дата подписания: 23.08.2017
Уникальный программный ключ:
c53e145e-8b20-45aa-9347-a5e4dbb90e8d

Томск 2018

Согласована на портале № 25088

Рассмотрена и одобрена на заседании кафедры
протокол № 6 от «15» 5 2018 г.

ЛИСТ СОГЛАСОВАНИЯ

Рабочая программа дисциплины составлена с учетом требований федерального государственного образовательного стандарта высшего образования (ФГОС ВО) по направлению подготовки (специальности) 10.05.02 Информационная безопасность телекоммуникационных систем, утвержденного 16.11.2016 года, рассмотрена и одобрена на заседании кафедры БИС «_» _____20___года, протокол №_____.

Разработчик:

Доцент каф. КИБЭВС _____А. С. Романов

Заведующий обеспечивающей каф. БИС _____Р. В. Мещеряков

Рабочая программа дисциплины согласована с факультетом и выпускающей кафедрой:

Декан ФБ _____Е. М. Давыдова

Заведующий выпускающей каф. БИС _____Р. В. Мещеряков

Эксперты:

Доцент каф. БИС _____О. О. Евсютин

Доцент каф. БИС _____А. Ю. Исхаков

Согласована на портале № 25088

1. Цели и задачи дисциплины

1.1. Цели дисциплины

Цель дисциплины «Языки программирования» - изучение основных теоретических и практических подходов в области разработки, реализации и использования языков и инструментальных средств программированя для решения профессиональных задач.

1.2. Задачи дисциплины

– В задачи дисциплины "Языки программирования" входит изучение следующих вопросов: основные сведения о характеристиках и свойствах языков программирования высокого уровня, принципы построения языков, средства описания данных; средства описания действий; абстрактные типы данных: инкапсуляция, спецификация, реализация, параметризация, классы и объекты; обработка файлов; обработка исключительных ситуаций; параллельная обработка; макропроцессоры и макрогенераторы; современные интегрированные среды разработки программ; графический интерфейс пользователя; отладчики; генераторы кода/приложений; библиотеки программ и классов; стандарты языков программирования, общая характеристика языков ассемблера: назначение, принципы построения и использования; структура языка, основные группы команд, операторы, средства взаимодействия с операционной системой.

2. Место дисциплины в структуре ОПОП

Дисциплина «Языки программирования» (Б1.Б.27) относится к блоку 1 (базовая часть).

Предшествующими дисциплинами, формирующими начальные знания, являются: Информатика, Основы программирования.

Последующими дисциплинами являются: Аппаратные средства телекоммуникационных систем, Системное программирование, Технологии и методы программирования.

3. Требования к результатам освоения дисциплины

Процесс изучения дисциплины направлен на формирование следующих компетенций:

– ОПК-5 способностью применять программные средства системного и прикладного назначения, языки, методы и инструментальные средства программирования для решения профессиональных задач;

В результате изучения дисциплины обучающийся должен:

– **знать** - основные теоретические и практические подходы в области разработки, реализации и использования языков и инструментальных средств программированя, применяемых для решения профессиональных задач

– **уметь** - применять языки и инструментальные средства программирования для разработки программных средств системного и прикладного назначения для решения профессиональных задач

– **владеть** - навыками использования современных программных средств системного и прикладного назначения и языков программирования высокого уровня для решения профессиональных задач

4. Объем дисциплины и виды учебной работы

Общая трудоемкость дисциплины составляет 4.0 зачетных единицы и представлена в таблице 4.1.

Таблица 4.1 – Трудоемкость дисциплины

Виды учебной деятельности	Всего часов	Семестры
		3 семестр
Аудиторные занятия (всего)	72	72
Лекции	36	36
Практические занятия	36	36
Из них в интерактивной форме	20	20
Самостоятельная работа (всего)	36	36

Согласована на портале № 25088

Проработка лекционного материала	25	25
Подготовка к практическим занятиям, семинарам	11	11
Всего (без экзамена)	108	108
Подготовка и сдача экзамена	36	36
Общая трудоемкость, ч	144	144
Зачетные Единицы	4.0	4.0

5. Содержание дисциплины

5.1. Разделы дисциплины и виды занятий

Разделы дисциплины и виды занятий приведены в таблице 5.1.

Таблица 5.1 – Разделы дисциплины и виды занятий

Названия разделов дисциплины	Лек., ч	Прак. зан., ч	Сам. раб., ч	Всего часов (без экзамена)	Формируемые компетенции
3 семестр					
1 Язык программирования. Классификация языков программирования	2	8	3	13	ОПК-5
2 Средства описания данных и действий	4	8	2	14	ОПК-5
3 Основные концепции объектно-ориентированного программирования. Абстрактные типы данных	6	8	4	18	ОПК-5
4 Обработка файлов	2	0	2	4	ОПК-5
5 Обработка исключительных ситуаций	2	4	4	10	ОПК-5
6 Параллельное программирование	4	0	2	6	ОПК-5
7 Макропроцессоры и макрогенераторы	2	0	2	4	ОПК-5
8 Современные интегрированные среды разработки программ; графический интерфейс пользователя	2	0	2	4	ОПК-5
9 Отладчики; генераторы кода/приложений	2	0	2	4	ОПК-5
10 Библиотеки программ и классов	2	0	2	4	ОПК-5
11 Языки низкоуровневого программирования	2	0	2	4	ОПК-5
12 Подпрограммы. Рекурсивный метод решения задач	2	4	4	10	ОПК-5
13 Основы функционального программирования	2	4	4	10	ОПК-5
14 Стандарты языков программирования и документирование программ	2	0	1	3	ОПК-5
Итого за семестр	36	36	36	108	
Итого	36	36	36	108	

Согласована на портале № 25088

5.2. Содержание разделов дисциплины (по лекциям)

Содержание разделов дисциплин (по лекциям) приведено в таблице 5.2.

Таблица 5.2 – Содержание разделов дисциплин (по лекциям)

Названия разделов	Содержание разделов дисциплины по лекциям	Трудоемкость, ч	Формируемые компетенции
3 семестр			
1 Язык программирования. Классификация языков программирования	Введение. Понятие языка программирования, программа, абстракция. Общие принципы построения и использования языков программирования; характеристики и свойства языков программирования. Пути развития и классификация языков программирования. Основные понятия языков программирования. Синтаксис и семантика языков программирования.	2	ОПК-5
	Итого	2	
2 Средства описания данных и действий	Средства описания данных. Типизация языка. Простые типы данных. Структурные типы данных. Структурные типы данных. Динамические структуры данных. Средства описания действий. Определение семантики средств описания действий. Элементы языков программирования. Выражения и операторы действия. Операторы управления. Блоки. Программы. Модули.	4	ОПК-5
	Итого	4	
3 Основные концепции объектно-ориентированного программирования. Абстрактные типы данных	Основные концепции объектно-ориентированного программирования. Абстрактные типы данных. Абстракции низкого и высокого уровней. Классы и объекты. Наследование, формы наследования, преимущества наследования. Инкапсуляция. Полиморфизм и его разновидности. Виртуальные функции. Отношения между классами, диаграммы классов. Преимущества и недостатки объектной модели.	6	ОПК-5
	Итого	6	
4 Обработка файлов	Файлы. Файлы прямого и последовательного доступа. Типы доступа. Обработка файлов. Открытие и закрытие файла. Запись в файл, чтение из файла.	2	ОПК-5
	Итого	2	
5 Обработка исключительных ситуаций	Обработка исключительных ситуаций. Простейшие способы обработки исключений. Обработка исключений при помощи меток. Структурный механизм обработки исключений. Восстановление вычислительного процесса. Определение семантики средств обработки исключений.	2	ОПК-5
	Итого	2	

Согласована на портале № 25088

6 Параллельное программирование	Параллельная обработка. Параллельное и распределенное программирование. Модели параллельного программирования. Процессы. Взаимное исключение. Синхронизация процессов. Передача сообщений. Сигналы, семафоры, рандеву.	4	ОПК-5
	Итого	4	
7 Макропроцессоры и макрогенераторы	Макропроцессоры и макрогенераторы.Принципы работы, способы реализации. Построение анализатора.	2	ОПК-5
	Итого	2	
8 Современные интегрированные среды разработки программ; графический интерфейс пользователя	Современные интегрированные среды разработки программ. Разработка графического интерфейс пользователя. Основные характеристики интегрированных средств разработки.	2	ОПК-5
	Итого	2	
9 Отладчики; генераторы кода/приложений	Отладчики. Генераторы кода/приложений. Режимы отладки, использование контрольных точек. Отладка программ без использования средств отладки. Основное назначение генераторов кода и приложений	2	ОПК-5
	Итого	2	
10 Библиотеки программ и классов	Библиотеки программ и классов. Принципы создания, правила ведения. Возможности пополнения и использования при собственной разработке.	2	ОПК-5
	Итого	2	
11 Языки низкоуровневого программирования	Общая характеристика низкоуровневых языков программирования: назначение, принципы построения и использования; структура языка, основные группы команд, операторы, средства взаимодействия с операционной системой. Трансляторы, компиляторы, интерпретаторы.	2	ОПК-5
	Итого	2	
12 Подпрограммы. Рекурсивный метод решения задач	Подпрограммы, реализация в различных языках программирования. Итерационный и рекурсивный методы решения задач. Типы рекурсии.	2	ОПК-5
	Итого	2	
13 Основы функционального программирования	Основы функционального программирования. Свойства и отличительные особенности функциональных языков программирования. Модель вычислений. Лямбда-исчисление. Аппликация, абстракция, редукция, эквивалентное преобразование. Ленивые и жадные вычисления.	2	ОПК-5
	Итого	2	
14 Стандарты языков программирования и документирование программ	Стандарты языков программирования и документирование программ.Стандарты языков программирования. Различия в стандартах языков программирования, стандартизация ISO. Стандарты документирования программ. ГОСТ ЕСПД. UML	2	ОПК-5

Согласована на портале № 25088

	диаграммы.		
	Итого	2	
Итого за семестр		36	

5.3. Разделы дисциплины и междисциплинарные связи с обеспечивающими (предыдущими) и обеспечиваемыми (последующими) дисциплинами

Разделы дисциплины и междисциплинарные связи с обеспечивающими (предыдущими) и обеспечиваемыми (последующими) дисциплинами представлены в таблице 5.3.

Таблица 5.3 – Разделы дисциплины и междисциплинарные связи

Наименование дисциплин	№ разделов данной дисциплины, для которых необходимо изучение обеспечивающих и обеспечиваемых дисциплин													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Предшествующие дисциплины														
1 Информатика	+	+						+						+
2 Основы программирования		+		+				+	+					
Последующие дисциплины														
1 Аппаратные средства телекоммуникационных систем					+									
2 Системное программирование		+	+	+		+				+	+			+
3 Технологии и методы программирования	+	+	+	+	+			+	+	+		+		

5.4. Соответствие компетенций, формируемых при изучении дисциплины, и видов занятий

Соответствие компетенций, формируемых при изучении дисциплины, и видов занятий представлено в таблице 5.4.

Таблица 5.4 – Соответствие компетенций, формируемых при изучении дисциплины, и видов занятий

Компетенции	Виды занятий			Формы контроля
	Лек.	Прак. зан.	Сам. раб.	
ОПК-5	+	+	+	Контрольная работа, Экзамен, Опрос на занятиях, Тест, Отчет по практическому занятию

6. Интерактивные методы и формы организации обучения

Технологии интерактивного обучения при разных формах занятий приведены в таблице 6.1.

Таблица 6.1 – Технологии интерактивного обучения при разных формах занятий

Методы	Интерактивные практические занятия, ч	Интерактивные лекции, ч	Всего, ч
3 семестр			
Презентации с использованием мультимедиа с обсуждением		10	10
IT-методы	10		10

Согласована на портале № 25088

Итого за семестр:	10	10	20
Итого	10	10	20

7. Лабораторные работы

Не предусмотрено РУП.

8. Практические занятия (семинары)

Наименование практических занятий (семинаров) приведено в таблице 8.1.

Таблица 8.1 – Наименование практических занятий (семинаров)

Названия разделов	Наименование практических занятий (семинаров)	Трудоемкость, ч	Формируемые компетенции
3 семестр			
1 Язык программирования. Классификация языков программирования	Анализ задачи. Абстракция программ и данных. Синтаксис языка программирования	4	ОПК-5
	Обработка текстов на естественном языке, синтаксические анализаторы	4	
	Итого	8	
2 Средства описания данных и действий	Вещественные числа. Ошибки при работе с вещественными числами	4	ОПК-5
	Рекурсия. Типы рекурсий	4	
	Итого	8	
3 Основные концепции объектно-ориентированного программирования. Абстрактные типы данных	Объектно-ориентированное программирование: модель предметной области и модель проектирования	4	ОПК-5
	Основные приемы объектно-ориентированного программирования	4	
	Итого	8	
5 Обработка исключительных ситуаций	Генерирование и обработка исключительных ситуаций	4	ОПК-5
	Итого	4	
12 Подпрограммы. Рекурсивный метод решения задач	Рекурсии. Типы рекурсий.	4	ОПК-5
	Итого	4	
13 Основы функционального программирования	Язык функционального программирования Haskell	4	ОПК-5
	Итого	4	
Итого за семестр		36	

9. Самостоятельная работа

Виды самостоятельной работы, трудоемкость и формируемые компетенции представлены в таблице 9.1.

Согласована на портале № 25088

Таблица 9.1 – Виды самостоятельной работы, трудоемкость и формируемые компетенции

Названия разделов	Виды самостоятельной работы	Трудоемкость, ч	Формируемые компетенции	Формы контроля
3 семестр				
1 Язык программирования. Классификация языков программирования	Подготовка к практическим занятиям, семинарам	2	ОПК-5	Контрольная работа, Опрос на занятиях, Отчет по практическому занятию, Тест, Экзамен
	Проработка лекционного материала	1		
	Итого	3		
2 Средства описания данных и действий	Подготовка к практическим занятиям, семинарам	1	ОПК-5	Контрольная работа, Опрос на занятиях, Отчет по практическому занятию, Тест, Экзамен
	Проработка лекционного материала	1		
	Итого	2		
3 Основные концепции объектно-ориентированного программирования. Абстрактные типы данных	Подготовка к практическим занятиям, семинарам	2	ОПК-5	Контрольная работа, Опрос на занятиях, Отчет по практическому занятию, Тест, Экзамен
	Проработка лекционного материала	2		
	Итого	4		
4 Обработка файлов	Проработка лекционного материала	2	ОПК-5	Контрольная работа, Опрос на занятиях, Тест, Экзамен
	Итого	2		
5 Обработка исключительных ситуаций	Подготовка к практическим занятиям, семинарам	2	ОПК-5	Контрольная работа, Опрос на занятиях, Отчет по практическому занятию, Тест, Экзамен
	Проработка лекционного материала	2		
	Итого	4		
6 Параллельное программирование	Проработка лекционного материала	2	ОПК-5	Контрольная работа, Опрос на занятиях, Тест, Экзамен
	Итого	2		
7 Макропроцессоры и макрогенераторы	Проработка лекционного материала	2	ОПК-5	Контрольная работа, Опрос на занятиях, Тест, Экзамен
	Итого	2		
8 Современные интегрированные среды разработки программ; графический интерфейс пользователя	Проработка лекционного материала	2	ОПК-5	Контрольная работа, Опрос на занятиях, Тест, Экзамен
	Итого	2		

Согласована на портале № 25088

9 Отладчики; генераторы кода/приложений	Проработка лекционного материала	2	ОПК-5	Контрольная работа, Опрос на занятиях, Тест, Экзамен
	Итого	2		
10 Библиотеки программ и классов	Проработка лекционного материала	2	ОПК-5	Контрольная работа, Опрос на занятиях, Тест, Экзамен
	Итого	2		
11 Языки низкоуровневого программирования	Проработка лекционного материала	2	ОПК-5	Контрольная работа, Опрос на занятиях, Тест, Экзамен
	Итого	2		
12 Подпрограммы. Рекурсивный метод решения задач	Подготовка к практическим занятиям, семинарам	2	ОПК-5	Контрольная работа, Опрос на занятиях, Отчет по практическому занятию, Тест, Экзамен
	Проработка лекционного материала	2		
	Итого	4		
13 Основы функционального программирования	Подготовка к практическим занятиям, семинарам	2	ОПК-5	Контрольная работа, Опрос на занятиях, Отчет по практическому занятию, Тест, Экзамен
	Проработка лекционного материала	2		
	Итого	4		
14 Стандарты языков программирования и документирование программ	Проработка лекционного материала	1	ОПК-5	Контрольная работа, Опрос на занятиях, Тест, Экзамен
	Итого	1		
Итого за семестр		36		
	Подготовка и сдача экзамена	36		Экзамен
Итого		72		

10. Курсовая работа (проект)

Не предусмотрено РУП.

11. Рейтинговая система для оценки успеваемости обучающихся

11.1. Балльные оценки для элементов контроля

Таблица 11.1 – Балльные оценки для элементов контроля

Элементы учебной деятельности	Максимальный балл на 1-ую КТ с начала семестра	Максимальный балл за период между 1КТ и 2КТ	Максимальный балл за период между 2КТ и на конец семестра	Всего за семестр
3 семестр				
Контрольная работа			10	10
Опрос на занятиях		5		5
Отчет по практическому занятию	15	15	15	45
Тест			10	10

Согласована на портале № 25088

Итого максимум за период	15	20	35	70
Экзамен				30
Нарастающим итогом	15	35	70	100

11.2. Пересчет баллов в оценки за контрольные точки

Пересчет баллов в оценки за контрольные точки представлен в таблице 11.2.

Таблица 11.2 – Пересчет баллов в оценки за контрольные точки

Баллы на дату контрольной точки	Оценка
≥ 90% от максимальной суммы баллов на дату КТ	5
От 70% до 89% от максимальной суммы баллов на дату КТ	4
От 60% до 69% от максимальной суммы баллов на дату КТ	3
< 60% от максимальной суммы баллов на дату КТ	2

11.3. Пересчет суммы баллов в традиционную и международную оценку

Пересчет суммы баллов в традиционную и международную оценку представлен в таблице 11.3.

Таблица 11.3 – Пересчет суммы баллов в традиционную и международную оценку

Оценка (ГОС)	Итоговая сумма баллов, учитывает успешно сданный экзамен	Оценка (ECTS)
5 (отлично) (зачтено)	90 - 100	A (отлично)
4 (хорошо) (зачтено)	85 - 89	B (очень хорошо)
	75 - 84	C (хорошо)
	70 - 74	D (удовлетворительно)
3 (удовлетворительно) (зачтено)	65 - 69	
	60 - 64	E (посредственно)
2 (неудовлетворительно) (не зачтено)	Ниже 60 баллов	F (неудовлетворительно)

12. Учебно-методическое и информационное обеспечение дисциплины

12.1. Основная литература

1. Формальные языки и компиляторы / Малявко А.А. - Новосиб.: НГТУ, 2014. - 431 с. - Режим доступа: http://znanium.com/bookread2.php?book=548152 [Электронный ресурс] - Режим доступа: http://znanium.com/bookread2.php?book=548152, дата обращения: 16.05.2018.

2. Ашарина, И.В. Объектно-ориентированное программирование в C++: лекции и упражнения [Электронный ресурс] : учеб. пособие — Электрон. дан. — Москва : Горячая линия-Телеком, 2012. — 320 с. — Режим доступа: https://e.lanbook.com/book/5115. — Загл. с экрана. [Электронный ресурс] - Режим доступа: https://e.lanbook.com/book/5115, дата обращения: 16.05.2018.

12.2. Дополнительная литература

1. Основы многопоточного и параллельного программирования: Учебное пособие / Карепова Е.Д. - Краснояр.:СФУ, 2016. - 356 с. - Режим доступа: http://znanium.com/bookread2.php?book=966962 [Электронный ресурс] - Режим доступа: http://znanium.com/bookread2.php?book=966962, дата обращения: 16.05.2018.

2. Объектно Ориентированное Программирование. Хорошая книга для Хороших Людей / Комлев Н.Ю. - М.:СОЛОН-Пр., 2015. - 298 с. - Режим доступа: http://znanium.com/bookread2.php?book=884394 [Электронный ресурс] - Режим доступа: http://znanium.com/bookread2.php?book=884394, дата обращения: 16.05.2018.

Согласована на портале № 25088

12.3. Учебно-методические пособия

12.3.1. Обязательные учебно-методические пособия

1. Языки программирования: Методические указания по лабораторным работам, практическим занятиям, самостоятельной и индивидуальной работе / Романов А. С. - 2018. 82 с. [Электронный ресурс] - Режим доступа: https://edu.tusur.ru/publications/7388, дата обращения: 16.05.2018.

12.3.2. Учебно-методические пособия для лиц с ограниченными возможностями здоровья и инвалидов

Учебно-методические материалы для самостоятельной и аудиторной работы обучающихся из числа лиц с ограниченными возможностями здоровья и инвалидов предоставляются в формах, адаптированных к ограничениям их здоровья и восприятия информации.

Для лиц с нарушениями зрения:

- в форме электронного документа;
- в печатной форме увеличенным шрифтом.

Для лиц с нарушениями слуха:

- в форме электронного документа;
- в печатной форме.

Для лиц с нарушениями опорно-двигательного аппарата:

- в форме электронного документа;
- в печатной форме.

12.4. Профессиональные базы данных и информационные справочные системы

1. 1. Научная электронная библиотека eLIBRARY.RU [Электронный ресурс]. - Режим доступа: https://elibrary.ru/defaultx.asp, свободный. - Загл. с экрана.

13. Материально-техническое обеспечение дисциплины и требуемое программное обеспечение

13.1. Общие требования к материально-техническому и программному обеспечению дисциплины

13.1.1. Материально-техническое и программное обеспечение для лекционных занятий

Для проведения занятий лекционного типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации используется учебная аудитория с количеством посадочных мест не менее 22-24, оборудованная доской и стандартной учебной мебелью. Имеются демонстрационное оборудование и учебно-наглядные пособия, обеспечивающие тематические иллюстрации по лекционным разделам дисциплины.

13.1.2.Материально-техническое и программное обеспечение для практических занятий

Аудитория информатики, технологий и методов программирования

учебная аудитория для проведения занятий лекционного типа, учебная аудитория для проведения занятий практического типа, помещение для курсового проектирования (выполнения курсовых работ), помещение для самостоятельной работы

634045, Томская область, г. Томск, ул. Красноармейская, д. 146, 408 ауд. Описание имеющегося оборудования:

- Интерактивная доска IQBoard 78" с ПО ActivInspire;
- Проектор ViewSonic PJD5154 DLP;
- Компьютеры класса не ниже M/B ASUS P5LD2 i945P / AMD A8 3.33 GHz / DDR-III DIMM 4096 Mb / Radeon R7 / 1 Gb Seagate (10 шт.);
- Комплект специализированной учебной мебели;
- Рабочее место преподавателя.

Программное обеспечение:

- Microsoft Windows 10
- Visual Studio
- Учебное сетевое ПО: Redmine, Putty
- Обучающее ПО: Visual studio, Git-bash

Согласована на портале № 25088

13.1.2. Материально-техническое и программное обеспечение для самостоятельной работы

Для самостоятельной работы используются учебные аудитории (компьютерные классы), расположенные по адресам:

- 634050, Томская область, г. Томск, Ленина проспект, д. 40, 233 ауд.;
- 634045, Томская область, г. Томск, ул. Красноармейская, д. 146, 201 ауд.;
- 634034, Томская область, г. Томск, Вершинина улица, д. 47, 126 ауд.;
- 634034, Томская область, г. Томск, Вершинина улица, д. 74, 207 ауд.

Состав оборудования:

- учебная мебель;
- компьютеры класса не ниже ПЭВМ INTEL Celeron D336 2.8ГГц. - 5 шт.;
- компьютеры подключены к сети «Интернет» и обеспечивают доступ в электронную информационно-образовательную среду университета.

Перечень программного обеспечения:

- Microsoft Windows;
- OpenOffice;
- Kaspersky Endpoint Security 10 для Windows;
- 7-Zip;
- Google Chrome.

13.2. Материально-техническое обеспечение дисциплины для лиц с ограниченными возможностями здоровья и инвалидов

Освоение дисциплины лицами с ограниченными возможностями здоровья и инвалидами осуществляется с использованием средств обучения общего и специального назначения.

При занятиях с обучающимися **с нарушениями слуха** предусмотрено использование звукоусиливающей аппаратуры, мультимедийных средств и других технических средств приема/передачи учебной информации в доступных формах, мобильной системы преподавания для обучающихся с инвалидностью, портативной индукционной системы. Учебная аудитория, в которой занимаются обучающиеся с нарушением слуха, оборудована компьютерной техникой, аудиотехникой, видеотехникой, электронной доской, мультимедийной системой.

При занятиях с обучающимися **с нарушениями зрениями** предусмотрено использование в лекционных и учебных аудиториях возможности просмотра удаленных объектов (например, текста на доске или слайда на экране) при помощи видеоувеличителей для комфортного просмотра.

При занятиях с обучающимися **с нарушениями опорно-двигательного аппарата** используются альтернативные устройства ввода информации и другие технические средства приема/передачи учебной информации в доступных формах, мобильной системы обучения для людей с инвалидностью.

14. Оценочные материалы и методические рекомендации по организации изучения дисциплины

14.1. Содержание оценочных материалов и методические рекомендации

Для оценки степени сформированности и уровня освоения закрепленных за дисциплиной компетенций используются оценочные материалы в составе:

14.1.1. Тестовые задания

Дан код:

```
int i1, i2;
int * const p1 = &i1;
const int* p2 = &i1;
const int* const p3 = &i1;
```

Какие из приведенных операций разрешено выполнять?

Согласована на портале № 25088

a) p1 = &i2;
b) *p1 = 5;
c) p2 = &i2;
d) *p2 = 5;
e) p3 = &i2; *p3 = 5;

Выберите вариант термина, которому соответствует определение:

«Механизм абстрагирования, который дает возможность описывать вычисление абстрактно, но в то же время, это описание может быть в любой момент переведено в форму понятную конкретному исполнителю (процессору)»

a) Синтаксис
b) Семантика
c) Язык программирования
d) Компилятор
e) Абстрактный класс

Какие из языков программирования относятся к высокоуровневым?

a) Python
b) C
c) C++
d) Assembler
e) Java

В каких из языков программирования применяет декларативный подход

a) C++
b) Haskell
c) Prolog
d) Java
e) Python

Синтаксис каких языков программировании наиболее близок к естественному языку?

a) Низкоуровневых
b) Объектно-ориентированных
c) Императивных
d) Декларативных
e) Все варианты верные

Как называется компоненты среды программирования, который позволяет делать трассировку программы, расставлять контрольные точки, отображать состояние регистров и флагов процессора?

a) Компилятор
b) Компоновщик
c) Отладчик
d) Профилировщик
e) Макропроцессор

Какие из приведенных языков являются интерпретируемыми?

a) Python
b) Ruby
c) C++
d) Go
e) Java

Выберите правильную последовательность выполнения процесса трансляции программы

Согласована на портале № 25088

a) Оптимизация кода -> синтаксический анализ ->семантический анализ -> генерация промежуточного кода -> лексический анализ -> генерация кода

b) Лексический анализ -> синтаксический анализ -> семантически анализ -> генерация промежуточного кода -> генерация кода -> оптимизация кода

c) Генерация промежуточного кода -> лексический анализ -> синтаксический анализ -> семантический анализ -> оптимизация кода -> генерация кода

d) Лексический анализ -> синтаксический анализ -> семантически анализ -> генерация промежуточного кода -> оптимизация кода -> генерация кода

e) Правильного ответа нет

Переменные, в которых связывание с памятью осуществляется при обработке операторов объявления переменных, а типы переменных связываются статически, называются…

a) Статическими
b) Стековыми
c) Явными динамическими
d) Неявными динамическими
e) Константами

Какие из приведенных типов данных языка C++ относятся к машинно-независимым?

a) int
b) char
c) double
d) bool
e) float

Ошибка при работе с вещественными числами, которая возникает когда результат невозможно представить в допустимом виде, называется…

a) Умножение ошибки
b) Исчезновение операнда
c) Потеря значимости
d) Переполнение
e) Погрешность

Для чего применяется операция стрелка -> при работе со структурами?

a) Для получения значения следующего поля структуры
b) Для доступа к элементу структуры по её имени
c) Для доступа к элементу структуры через указатель
d) Для преобразования структуры в массив.
e) Верного ответа нет

Какие из приведенных операторов можно считать основными в любом языке программирования?

a) Операторы цикла
b) Условные операторы
c) Оператор безусловного перехода
d) Оператор присваивания
e) Операторы вызова подпрограмм

Чему будет равно значение переменной counter после выполнения цикла?

```
int counter=0;
for(;;) {counter++;}
cout << to_string(counter);
```

a) В коде допущены ошибки, он не будет работать

Согласована на портале № 25088

b) Значение counter будет равно 0, т.к. в цикле не задано условие останова.

c) Значение counter будет равно 1, т.к. в цикле не задано условие останова, но цикл выполняется всегда хотя бы один раз.

d) Это бесконечный цикл, поэтому counter будет увеличиваться, пока не произойдет ошибка переполнение переменной

e) Это бесконечный цикл, вывода значения переменной не произойдет

К какому виду рекурсии можно отнести функцию ниже

```
int f (int i) {
f(f(i-1)+f(i-2));
}
```

a) Линейная
b) Повторная
c) Взаимная
d) Удаленная
e) Каскадная

```
void func (int *x, int *y)
{
int temp = *x;
*x = *y;
*y = temp;
}
…
int a=10;
int b=20;
func(&a,&b);
```

Чему будут равны переменные a и b?

a) a=10 b=20
b) a=20 b=10
c) a=10 b=10
d) a=20 b=20
e) нельзя использовать и указатели и ссылки одновременно

Класс, который содержит только объявление методов и статические константные поля называется…

a) Классом-утилитой
b) Абстрактным
c) Интерфейсом
d) Эффективным
e) Полиморфным

Операция, которая дает доступ к содержанию объекта по частям, в строго определенном порядке называется

a) Конструктор
b) Деструктор
c) Модификатор
d) Итератор
e) Селектор

Определить, не компилируя, что напечатает программа.

```
class A {
public:
```

Согласована на портале № 25088

```
A() { cout << "A() "; }
~A() { cout << "~A() ";}
};

class B : public A {
public:
B() { cout << "B() "; }
~B() { cout << "~B() "; }
};
int main()
{
A * pA = new B;
delete pA;
return EXIT_SUCCESS;
}
```

a) A() B() ~A() ~B()
b) A() B() ~B() ~A()
c) A() B() ~A()
d) A() B() ~B()

Расположите типы связности от худшей к лучшей:

1. Информационная
2. Случайная
3. Последовательная
4. Логическая
5. Коммуникационная
6. Процедурная
7. Функциональная
8. Временная

a) 3 8 4 1 7 2 6 5
b) 2 4 8 6 5 3 7 1
c) 8 1 3 4 6 7 5 2
d) 2 4 1 3 8 6 7 5
e) 4 6 7 8 5 3 2 1

Для чего нужен оператор const_cast в языке C++?

a) преобразует тип операнда с пометкой const в аналогичный тип, но без пометки const.
b) это продвинутый аналог оператора const для C++, который не позволяет делать указатель на переменную.
c) это оператор, позволяющий сделать константный указатель на переменную в C++.
d) это оператор, позволяющий сделать указатель на константу в C++.
e) ни один из ответов не является верным

Дан фрагмент кода

```
class B {
int a;
public: int b;
protected: int c;
private: int d;
};

class D: public B {
int fa() {return a;} //1
```

Согласована на портале № 25088

```
int fb() {return b;} //2
int fc() {return c;} //3
int fd() {return d;} //4
}
```

В каких строках (1,2,3,4) допущена ошибка?

a) 1 строка
b) 2 строка
c) 3 строка
d) 4 строка
e) во всех четырех строках

Что обозначается ключевым словом catch в C++?

a) контролируемый блок
b) блок обработки исключения
c) секция-ловушка
d) генератор исключения
e) обработчик прерывания

Множественное наследование –это…

a) когда у каждого производного класса не более одного родительского класса
b) иерархическая система классов, каждый уровень которой построен по принципу один базовый класс – один производный класс
c) когда у двух классов есть общий производный класс
d) когда есть полиморфный класс
e) все ответы верны

Определить, не компилируя, что напечатает программа.

```
class A {
public:
A() { cout << "A() "; }
virtual ~A() { cout << "~A() ";}
};

class B : public A {
public:
B() { cout << "B() "; }
~B() { cout << "~B() "; }
};
int main()
{
A * pA = new B;
delete pA;
return EXIT_SUCCESS;
}
```

a) A() B() ~A() ~B()
b) A() B() ~B() ~A()
c) A() B() ~A()
d) A() B() ~B()

Какой оператор используется для перехвата исключения?

a) try
b) catch

Согласована на портале № 25088

c) throw
d) finally
e) help

Какой оператор используется для инициализации исключения?

a) try
b) catch
c) throw
d) finally
e) help

Укажите правильное объявление виртуального метода, который принимает одно целочисленное значение и возвращает void.

a) virtual void SomeFunction (int x);
b) void SomeFunction (int x) virtual;
c) virtual SomeFunction (int x);
d) virtual void SomeFunction (int * x);
e) нет правильного ответа

Принцип объектно-ориентированного программирования, заключается в объединении атрибутов и методов объекта с целью обеспечения сохранности данных, называется...

a) наследование
b) абстрагирование
c) инкапсуляция
d) типизация
e) инициализация

Какой из вариантов записи абстрактного класса в C ++ является правильным?

a) abstract class A {virtual int f () = 0;};
b) class A {virtual int f () = 0;};
c) class A {virtual int f () = 0;} abstract;
d) class A {virtual int f ();};
e) все ответы верны

Выберите верное утверждение о деструктор класса в C ++

a) Деструктор принимает в качестве параметра адрес того объекта, который нужно уничтожить.
b) Деструктор принимает в качестве параметра указатель this.
c) Деструктор не содержит параметров.
d) Деструктор принимает в качестве параметра тот объект, который нужно уничтожить.
e) Деструктор принимает в качестве параметра тот объект, который нужно уничтожить.

14.1.2. Экзаменационные вопросы

1. Язык программирования. Общие принципы построения и использования язы¬ков программирования.
2. Стандарты языков программирования.
3. Лямбда-исчисление. Аппликация, абстракция, редукция, преобразование.
4. Списки и функциональные выражения в функциональных языках программирования.
5. Механизмы и средства взаимодействия программы с операционной системой.
6. Функциональное программирование. Основные положения. Основные отличия от других типов языков программирования.
7. Классификация языков программирования. Близость языков программирования к естественному языку.
8. Унификация и хорновский клоз в логических языках программирования.
9. Модель вычислений функциональных языков программирования.

Согласована на портале № 25088

10. Языки программирования низкого уровня.

11. Средства разработки графического интерфейса пользователя. Эргономические свойства человеко-машинного интерфейса.

12. Процедурные языки программирования. Основные отличия от других типов языков.

13. Обоснование выбора языка программирования.

14. Перегрузка в языках программирования.

15. Логические языки программирования. Основные положения и понятия. Отличия от других типов языков программирования.

16. Объектно-ориентированные языки программирования. Основные отличия от других концепций языков программирования.

17. Полиморфизм в объектно-ориентированном программировании. Виртуальные функции. Таблицы виртуальных функций.

18. Структура языка программирования. Синтаксис и семантика языков программирования. Расширенная форма Бэкуса-Наура.

19. Наследование в объектно-ориентированном программировании. Множественное наследование. Проблемы множественного наследования.

20. Инкапсуляция в объектно-ориентированном программировании. Контроль доступа.

21. Понятие класса и объекта в объектно-ориентированном программировании. Атрибуты, методы, конструктор и деструктор, статические члены класса.

22. Диаграммы классов UML. Основные элементы и обозначения.

23. Данные. Средства описания данных. Типизация языка.

24. Преобразование типов. Контроль соответствия типов данных.

25. Объектно-ориентированное программирование. Основные концепции объектно-ориентированного программирования.

26. Современные интегриро¬ванные среды разработки программ. Основные компоненты среды программирования.

27. Декомпозиция программ.

28. Трансляторы. Интерпретация и компиляция.

29. Макропроцессоры и макрогенераторы.

30. Потоки и процессы. Сходства и различия.

31. Мониторы и защищаемые переменные в параллельном программировании.

32. Семафоры в параллельном программировании. Типы семафоров.

33. Отладчики. Генераторы кода и приложений.

34. Параллельная обработка данных и параллелизм. Параллельное и распределенное программирование.

35. Основные проблемы параллельного и распределенного программирования.

36. Оценка максимально возможного параллелизма.

37. Основные модели параллельного программирования.

38. Оптимизатор. Основные функции оптимизатора.

39. Обработка исключительных ситуаций. Иерархия исключительных ситуаций. Виды исключительных ситуаций.

40. Операторы обработки исключительных ситуаций в различных языках программирования.

41. Схема обработки исключительных ситуаций Б. Мейера.

42. Элементарные типы данных.

43. Перегрузка данных, операторов, методов.

44. Составные типы данных.

45. Механизмы логического вывода. Прямая и обратная цепочки рассуждений.

46. Пространство имен, область видимости, время жизни переменных.

47. Ошибки при работе с вещественными числами. Смешанная арифметика.

48. Операторы выбора и условные операторы.

49. Вещественные числа. Способы представления. Операции над вещественными числами.

50. Оператор присваивания. Операторы цикла.

51. Распределение памяти при выполнении программы.

Согласована на портале № 25088

52. Куча. Менеджер кучи. Фрагментация динамической памяти.

53. Концепция виртуальной памяти. Страничная организация памяти.

54. Сегментный принцип организации памяти. Сегментация памяти.

55. Указатели. Операции над указателями. Типизированные и нетипизированные указатели.

56. Динамические структуры данных. Реализация динамических структур данных с помощью указателей.

57. Библиотеки программ и классов. Статические и динамические библиотеки. Критерии проектирования библиотек.

58. Подпрограммы. Формальные и фактические параметры подпрограмм.

59. Передача параметров подпрограмме.

60. Программный стек и его изменение.

61. Рекурсивный и итерационный методы решения задач. Виды рекурсий.

62. Общая характеристика языков ассемблера: назна¬чение, принципы построения и использования; структура языка.

63. Сериализация и десериализация. Методы сериализации объектов в базу данных.

64. Динамическая диспетчеризация.

65. Родовые (настраиваемые) сегменты.

66. Шаблоны.

67. Вариантные записи.

68. Средства описания действий над данными. Операторы, выражения, модули, блоки в языках программирования.

69. Ленивые и жадные вычисления в процедурном и функциональном программировании.

70. Языки высокого уровня.

71. Показатели качества программных средств.

72. Отношения между классами в объектно-ориентированном программировании.

73. Файлы, обработка файлов. Типы доступа к файлам.

74. Абстрактные типы данных: инкапсуляция, спецификация, реализация, параметризация.

75. Основные группы команд, опера¬торы, средства взаимодействия с операционной системой в языках ассемблера.

14.1.3. Темы контрольных работ

Основные концепции языков программирования

14.1.4. Темы опросов на занятиях

Введение. Понятие языка программирования, программа, абстракция. Общие принципы построения и использования языков программирования; характеристики и свойства языков программирования. Пути развития и классификация языков программирования. Основные понятия языков программирования. Синтаксис и семантика языков программирования.

Средства описания данных. Типизация языка. Простые типы данных. Структурные типы данных. Структурные типы данных. Динамические структуры данных. Средства описания действий. Определение семантики средств описания действий. Элементы языков программирования. Выражения и операторы действия. Операторы управления. Блоки. Программы. Модули.

Основные концепции объектно-ориентированного программирования. Абстрактные типы данных. Абстракции низкого и высокого уровней. Классы и объекты. Наследование, формы наследования, преимущества наследования. Инкапсуляция. Полиморфизм и его разновидности. Виртуальные функции. Отношения между классами, диаграммы классов. Преимущества и недостатки объектной модели.

Файлы. Файлы прямого и последовательного доступа. Типы доступа. Обработка файлов. Открытие и закрытие файла. Запись в файл, чтение из файла.

Обработка исключительных ситуаций. Простейшие способы обработки исключений. Обработка исключений при помощи меток. Структурный механизм обработки исключений. Восстановление вычислительного процесса. Определение семантики средств обработки исключений.

Параллельная обработка. Параллельное и распределенное программирование. Модели параллельного программирования. Процессы. Взаимное исключение. Синхронизация процессов. Передача сообщений. Сигналы, семафоры, рандеву.

Макропроцессоры и макрогенераторы.

Согласована на портале № 25088

Принципы работы, способы реализации. Построение анализатора.

Современные интегрированные среды разработки программ. Разработка графического интерфейс пользователя. Основные характеристики интегрированных средств разработки.

Отладчики. Генераторы кода/приложений. Режимы отладки, использование контрольных точек. Отладка программ без использования средств отладки. Основное назначение генераторов кода и приложений

Библиотеки программ и классов. Принципы создания, правила ведения. Возможности пополнения и использования при собственной разработке.

Общая характеристика низкоуровневых языков программирования: назначение, принципы построения и использования; структура языка, основные группы команд, операторы, средства взаимодействия с операционной системой. Трансляторы, компиляторы, интерпретаторы.

Подпрограммы, реализация в различных языках программирования. Итерационный и рекурсивный методы решения задач. Типы рекурсии.

Основы функционального программирования. Свойства и отличительные особенности функциональных языков программирования. Модель вычислений. Лямбда-исчисление. Аппликация, абстракция, редукция, эквивалентное преобразование. Ленивые и жадные вычисления.

Стандарты языков программирования и документирование программ.

Стандарты языков программирования. Различия в стандартах языков программирования, стандартизация ISO. Стандарты документирования программ. ГОСТ ЕСПД. UML диаграммы.

14.1.5. Вопросы для подготовки к практическим занятиям, семинарам

Анализ задачи. Абстракция программ и данных. Синтаксис языка программирования
Обработка текстов на естественном языке, синтаксические анализаторы
Вещественные числа. Ошибки при работе с вещественными числами
Рекурсия. Типы рекурсий
Генерирование и обработка исключительных ситуаций
Объектно-ориентированное программирование:
модель предметной области и модель проектирования
Основные приемы объектно-ориентированного программирования
Рекурсии. Типы рекурсий.
Язык функционального программирования Haskell

14.2. Требования к оценочным материалам для лиц с ограниченными возможностями здоровья и инвалидов

Для лиц с ограниченными возможностями здоровья и инвалидов предусмотрены дополнительные оценочные материалы, перечень которых указан в таблице 14.

Таблица 14 – Дополнительные материалы оценивания для лиц с ограниченными возможностями здоровья и инвалидов

Категории обучающихся	Виды дополнительных оценочных материалов	Формы контроля и оценки результатов обучения
С нарушениями слуха	Тесты, письменные самостоятельные работы, вопросы к зачету, контрольные работы	Преимущественно письменная проверка
С нарушениями зрения	Собеседование по вопросам к зачету, опрос по терминам	Преимущественно устная проверка (индивидуально)
С нарушениями опорно-двигательного аппарата	Решение дистанционных тестов, контрольные работы, письменные самостоятельные работы, вопросы к зачету	Преимущественно дистанционными методами
С ограничениями по общемедицинским показаниям	Тесты, письменные самостоятельные работы, вопросы к зачету, контрольные работы, устные ответы	Преимущественно проверка методами исходя из состояния обучающегося на момент проверки

Согласована на портале № 25088

14.3. Методические рекомендации по оценочным материалам для лиц с ограниченными возможностями здоровья и инвалидов

Для лиц с ограниченными возможностями здоровья и инвалидов предусматривается доступная форма предоставления заданий оценочных средств, а именно:

- в печатной форме;
- в печатной форме с увеличенным шрифтом;
- в форме электронного документа;
- методом чтения ассистентом задания вслух;
- предоставление задания с использованием сурдоперевода.

Лицам с ограниченными возможностями здоровья и инвалидам увеличивается время на подготовку ответов на контрольные вопросы. Для таких обучающихся предусматривается доступная форма предоставления ответов на задания, а именно:

- письменно на бумаге;
- набор ответов на компьютере;
- набор ответов с использованием услуг ассистента;
- представление ответов устно.

Процедура оценивания результатов обучения лиц с ограниченными возможностями здоровья и инвалидов по дисциплине предусматривает предоставление информации в формах, адаптированных к ограничениям их здоровья и восприятия информации:

Для лиц с нарушениями зрения:

- в форме электронного документа;
- в печатной форме увеличенным шрифтом.

Для лиц с нарушениями слуха:

- в форме электронного документа;
- в печатной форме.

Для лиц с нарушениями опорно-двигательного аппарата:

- в форме электронного документа;
- в печатной форме.

При необходимости для лиц с ограниченными возможностями здоровья и инвалидов процедура оценивания результатов обучения может проводиться в несколько этапов.

Согласована на портале № 25088