

Министерство науки и высшего образования РФ  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

**СОГЛАСОВАНО**

**Заведующий кафедрой**

**Базовая кафедра**  
**вычислительных и**  
**информационных технологий**  
**(ВиИТ\_ФМиИ)**

наименование кафедры

подпись, инициалы, фамилия

«\_\_\_» \_\_\_\_\_ 20\_\_ г.

институт, реализующий ОП ВО

**УТВЕРЖДАЮ**

**Заведующий кафедрой**

**Базовая кафедра вычислительных**  
**и информационных технологий**  
**(ВиИТ\_ФМиИ)**

наименование кафедры

**д.ф.-м.н., профессор Шайдуров**  
**В.В.**

подпись, инициалы, фамилия

«\_\_\_» \_\_\_\_\_ 20\_\_ г.

институт, реализующий дисциплину

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ**  
**ПРОГРАММИРОВАНИЕ**

Дисциплина Б1.О.26 Программирование

Направление подготовки /  
специальность 02.03.01 Математика и компьютерные науки  
Профиль 02.03.01.31 Математическое и  
компьютерное моделирование

Направленность  
(профиль)

Форма обучения очная

Год набора 2019

Красноярск 2021

## **РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ**

составлена в соответствии с Федеральным государственным образовательным стандартом высшего образования с учетом профессиональных стандартов по укрупненной группе

020000 «КОМПЬЮТЕРНЫЕ И ИНФОРМАЦИОННЫЕ НАУКИ»

Направление подготовки /специальность (профиль/специализация)

Направление 02.03.01 Математика и компьютерные науки Профиль

02.03.01.31 Математическое и компьютерное моделирование

Программу  
составили

к.ф.н., доцент, Олейников Б.В.

## **1 Цели и задачи изучения дисциплины**

### **1.1 Цель преподавания дисциплины**

Основная цель преподавания дисциплины - изучение и применение современных парадигм, методологий, сред, языков программирования, алгоритмов и структур данных, включая сложные динамические структуры данных для решения научных и практических задач различных предметных областей.

### **1.2 Задачи изучения дисциплины**

Основными задачами изучения дисциплин являются:

- изучение процедурного программирования и основных алгоритмов на основе базового языка программирования (Object Pascal, PascalABC.Net), а также статических структур данных и их практическое применение при решении математических и прикладных задач;
- изучение сложных динамических структур данных (линейных и нелинейных), алгоритмов работы с ними и их программирование на основе использования базового языка программирования (Object Pascal, PascalABC.Net);
- изучение объектно-ориентированного и визуального программирования и их практическое применение на основе использования базового языка программирования (Object Pascal), базовой RAD-среды (Delphi) и языка объектно-ориентированного проектирования UML с использованием базовой среды (Together или одной из свободно распространяемых сред, например, Star UML).

1.3 Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения образовательной программы

<b>ОПК-4:Способен находить, анализировать, реализовывать программно и использовать на практике математические алгоритмы, в том числе с применением современных вычислительных систем</b>
<b>ОПК-4.1:Находит и анализирует математические алгоритмы для решения практических задач</b>
<b>ОПК-4.2:Реализовывает программно и использует на практике математические алгоритмы, в том числе с применением современных вычислительных систем</b>
<b>ОПК-5:Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности</b>
<b>ОПК-5.1:Использует электронные библиотечные системы, национальные и международные базы данных для поиска необходимой научной литературы</b>
<b>ОПК-5.2:Применяет современное программное обеспечение для решения</b>

#### 1.4 Место дисциплины (модуля) в структуре образовательной программы

Дисциплина состоит из 3-х семестровых частей, изучаемых в 1, 2 и 3 семестрах. Для успешного освоения 1-ой части желательно знакомство с основами информатики в рамках школьного курса информатики, а также знание понятий и алгоритмов, изучаемых в школьном курсе математики (алгебра и геометрия), основ теории множеств. Изучение 2-ой части опирается на материал 1-ой части, а также на начальные сведения из теории графов, общей теории структур. Изучение 3-ей части опирается на материал первых двух частей, а также на теорию проектирования общих информационных систем, изучаемой в рамках программы подготовки бакалавра.

Дисциплина «Программирование» является основной для изучения многих дисциплин профессионального цикла профиля, включая «Численные методы», «Базы данных», «Архитектура компьютеров», «Математическое моделирование» и др., а также многих дисциплин вариативной части, связанных с программированием задач конкретной тематической области.

#### 1.5 Особенности реализации дисциплины

Язык реализации дисциплины Русский.

Дисциплина (модуль) реализуется с применением ЭО и ДОТ

<https://e.sfu-kras.ru/course/view.php?id=2263>

<https://e.sfu-kras.ru/course/view.php?id=2261>

<https://e.sfu-kras.ru/course/view.php?id=2260>

<http://study.sfu-kras.ru/course/view.php?id=6>

<http://study.sfu-kras.ru/course/view.php?id=121>

## 2. Объем дисциплины (модуля)

Вид учебной работы	Всего, зачетных единиц (акад.час)	Семестр		
		1	2	3
<b>Общая трудоемкость дисциплины</b>	<b>15 (540)</b>	<b>4,5 (162)</b>	<b>4,5 (162)</b>	<b>6 (216)</b>
<b>Контактная работа с преподавателем:</b>	<b>7,36 (265)</b>	<b>2,5 (90)</b>	<b>2,36 (85)</b>	<b>2,5 (90)</b>
занятия лекционного типа	1,47 (53)	0,5 (18)	0,47 (17)	0,5 (18)
занятия семинарского типа				
в том числе: семинары				
практические занятия	5,89 (212)	2 (72)	1,89 (68)	2 (72)
практикумы				
лабораторные работы				
другие виды контактной работы				
в том числе: групповые консультации				
индивидуальные консультации				
иная внеаудиторная контактная работа:				
групповые занятия				
индивидуальные занятия				
<b>Самостоятельная работа обучающихся:</b>	<b>5,64 (203)</b>	<b>2 (72)</b>	<b>1,14 (41)</b>	<b>2,5 (90)</b>
изучение теоретического курса (ТО)				
расчетно-графические задания, задачи (РГЗ)				
реферат, эссе (Р)				
курсовое проектирование (КП)	Нет	Нет	Нет	Нет
курсовая работа (КР)	Нет	Нет	Нет	Нет
<b>Промежуточная аттестация (Зачёт) (Экзамен)</b>	<b>2 (72)</b>		<b>1 (36)</b>	<b>1 (36)</b>

### 3 Содержание дисциплины (модуля)

#### 3.1 Разделы дисциплины и виды занятий (тематический план занятий)

№ п/п	Модули, темы (разделы) дисциплины	Занятия лекционного типа (акад. час)	Занятия семинарского типа		Самостоятельная работа, (акад. час)	Формируемые компетенции
			Семинары и/или Практические занятия (акад. час)	Лабораторные работы и/или Практикумы (акад. час)		
1	2	3	4	5	6	7
1	1	10	36	0	36	
2	2	8	36	0	36	
3	3	8	32	0	20	
4	4	9	36	0	21	
5	5	10	40	0	45	
6	6	8	32	0	45	
Всего		53	212	0	203	

#### 3.2 Занятия лекционного типа

№ п/п	№ раздела дисциплины	Наименование занятий	Объем в акад. часах		
			Всего	в том числе, в инновационной форме	в том числе, в электронной форме
1	1	1.1. Введение. Базовые понятия (информация, алгоритм, структура данных, язык, программа, компиляция, этапы компиляции, операционная система, отладка, тестирование и др.). Основные сведения	2	0	0
2	1	1.2. Типизация данных. Основные типы данных и средства работы с ними. Основные операторы	2	0	0

3	1	1.3.Проектируемые типы для числовых данных. Основные алгоритмы для работы с ними	2	0	0
4	1	1.4.Основы и инструменты структурного программирования. Процедурные типы. Рекуррентность и рекурсия. Реализация в языке	2	0	0
5	1	1.5.Мат. индукция, рекурсия и итерация. Фракталы	2	0	0
6	2	2.1.Нечисловые типы данных (строки, множества) и инструменты работы с ними	2	0	0
7	2	2.2.Комбинированный тип данных. Файловый тип данных. Основные инструменты по работе с ними	2	0	0
8	2	2.3.Расширенная работа с файлами. Типизированные константы. Преобразование типов	2	0	0
9	2	2.4.Обработка исключительных ситуаций. Создание отдельно компилируемых подпрограмм.	2	0	0
10	3	3.1.Уровни представления данных. Структуры данных. Представление структур. Графы	2	0	0

11	3	3.2. Статическая и динамическая память. Адресация памяти в архитектуре Intel 8086/88. Средства работы с динамической памятью в базовом языке	2	0	0
12	3	3.3. Линейные динамические структуры данных. Работа с ними. Области применения	2	0	0
13	3	3.4. Большие числа. Их представление линейными динамическими структурами. Реализация основных операций над ними	2	0	0
14	4	4.1. Виды нелинейных динамических структур. Бинарные деревья. Операции над ними. Прошитые деревья. Области применения	2	0	0
15	4	4.2. Поиск на деревьях. Сбалансированные деревья. Построение и использование AVL-деревьев	2	0	0
16	4	4.3. «Цветная» балансировка деревьев. RB- и AA-деревья. Построение и использование	2	0	0
17	4	4.4. N-арные деревья. Способы представления. B-деревья. Основные операции. Крупномасштабные N-арные деревья – как основа представления данных в СУБД.	3	0	0

18	5	5.1. Введение. Модели мира и парадигмы программирования. ООМ. История. Среды. Языки	2	0	0
19	5	5.2. ООП. Основные понятия. Концепция. Абстракция. Инкапсуляция. Поля, методы. Конструктор, деструктор. Свойства. Классификация свойств	2	0	0
20	5	5.3. Наследование. Модуль. Области видимости. События. Перекрытие методов. Виды методов. Полиморфизм. Класс и объект «изнутри».	2	0	0
21	5	5.4. Обработка сообщений. Перегрузка методов. Событие. Делегирование событий. Работа с классами. Методы и операторы для работы с классами. Библиотека классов VCL.	2	0	0
22	5	5.5. Контейнер. Итератор. Интерфейсы. Стандартные контейнеры в библиотеке VCL и других.	2	0	0
23	6	6.1. Обобщенное программирование. Анонимные методы	2	0	0
24	6	6.2. Визуальное программирование в среде Delphi. Создание графического интерфейса пользователя. Использование файлов и изображений	2	0	0

25	6	6.3. Жизненный цикл ПО. ООМ. Язык UML – как инструмент ООМ. История, понятия, структура языка UML	2	0	0
26	6	6.4. Основные диаграммы языка UML. Назначение. Нотация. Примеры. Среды поддержки языка UML	2	0	0
Всего			52	0	0

### 3.3 Занятия семинарского типа

№ п/п	№ раздела дисциплины	Наименование занятий	Объем в акад. часах		
			Всего	в том числе, в инновационной форме	в том числе, в электронной форме
1	1	1.1. Структура программы. Операторы ввода вывода. Числовые типы данных. Логические и арифметические выражения. Блочный оператор. Условный оператор. (Написание программ)	2	0	0
2	1	Структура программы. Операторы ввода вывода. Числовые типы данных. Логические и арифметические выражения. Блочный оператор. Условный оператор. (Разбор алгоритмов)	2	0	0
3	1	1.2. Основные типы простых данных. Основные операции. Операторы цикла. Операторы выбора. (Написание программ)	4	0	0
4	1	1.2. Основные типы простых данных. Основные операции. Операторы цикла. Операторы выбора. (Разбор алгоритмов)	4	0	0

5	1	1.3. Проектируемые типы для числовых данных. Массивы. Основные алгоритмы для работы с ними. Сортировки. (Написание программ)	6	0	0
6	1	1.3. Проектируемые типы для числовых данных. Массивы. Основные алгоритмы для работы с ними. Сортировки. (Разбор алгоритмов)	6	0	0
7	1	1.4. Организация подпрограмм. Процедуры и функции. Процедурные типы. Прямая и косвенная рекурсия. Построение рекурсивных подпрограмм	4	0	0
8	1	1.4. Организация подпрограмм. Процедуры и функции. Процедурные типы. Прямая и косвенная рекурсия. Построение рекурсивных подпрограмм	4	0	0
9	1	1.5. Построение геометрических фракталов. Примеры использования генераторов фракталов. (Написание программ)	2	0	0
10	1	1.5. Построение геометрических фракталов. Примеры использования генераторов фракталов (Разбор алгоритмов)	2	0	0
11	2	2.1. Нечисловые типы данных (строки, множества) и инструменты работы с ними. (Написание программ)	4	0	0
12	2	2.1. Нечисловые типы данных (строки, множества) и инструменты работы с ними. (Разбор алгоритмов)	4	0	0

13	2	2.2. Комбинированный тип данных. Файловый тип данных. Основные инструменты по работе с ними. Написание программ.	6	0	0
14	2	2.2. Комбинированный тип данных. Файловый тип данных. Основные инструменты по работе с ними. Разбор алгоритмов.	6	0	0
15	2	2.3. Расширенная работа с файлами. Типизированные константы. Преобразование типов. (Написание программ)	6	0	0
16	2	2.3. Расширенная работа с файлами. Типизированные константы. Преобразование типов. (Разбор алгоритмов)	6	0	0
17	2	2.4. Обработка исключительных ситуаций. Создание отдельно компилируемых подпрограмм. (Написание программ)	2	0	0
18	2	2.4. Обработка исключительных ситуаций. Создание отдельно компилируемых подпрограмм. (Разбор алгоритмов)	2	0	0
19	3	3.1. Представление в программе графовых структур. Решение задач (остовное дерево, кратчайший путь). Написание программ.	2	0	0
20	3	3.1. Представление в программе графовых структур. Решение задач (остовное дерево, кратчайший путь). Разбор алгоритмов	2	0	0

21	3	3.2. Использование указателей и работа с динамической памятью. Использование основных стандартных процедур и функций по работе с динамической памятью в базовом языке. (Написание программ)	4	0	0
22	3	3.2. Использование указателей и работа с динамической памятью. Использование основных стандартных процедур и функций по работе с динамической памятью в базовом языке. (Разбор алгоритмов)	4	0	0
23	3	3.3. Организация линейных динамических структур данных (стек, очередь, дек, кольцо). Работа с ними. Написание программ.	6	0	0
24	3	3.3. Организация линейных динамических структур данных (стек, очередь, дек, кольцо). Работа с ними. Разбор алгоритмов.	6	0	0
25	3	3.4. Представление больших чисел линейными динамическими структурами. Реализация основных арифметических операций на них. Написание программ.	4	0	0
26	3	3.4. Представление больших чисел линейными динамическими структурами. Реализация основных арифметических операций на них. Разбор алгоритмов.	4	0	0

27	4	4.1. Построение бинарных деревьев и реализация основных операций над ними. Создание прошитых деревьев. Написание программ.	6	0	0
28	4	4.1. Построение бинарных деревьев и реализация основных операций над ними. Создание прошитых деревьев. Разбор алгоритмов.	6	0	0
29	4	4.2. Балансировка деревьев вращением. Построение и использование AVL-деревьев. Написание программ.	4	0	0
30	4	4.2. Балансировка деревьев вращением. Построение и использование AVL-деревьев. Разбор алгоритмов.	4	0	0
31	4	4.3. «Цветная» балансировка деревьев. Построение и использование RB- и AA-деревьев. Написание программ.	4	0	0
32	4	4.3. «Цветная» балансировка деревьев. Построение и использование RB- и AA-деревьев. Разбор алгоритмов.	4	0	0
33	4	4.4. Создание N-арных деревьев. Создание B-деревьев. Реализация основных операций. Написание программ.	4	0	0
34	4	4.4. Создание N-арных деревьев. Создание B-деревьев. Реализация основных операций. Разбор алгоритмов.	4	0	0
35	5	5.1. Анатомия класса и объекта. Примеры построения. Написание программ.	2	0	0

36	5	5.1. Анатомия класса и объекта. Примеры построения. Разбор алгоритмов.	2	0	0
37	5	5.2. Реализация инкапсуляции. Представление свойств в классе. Написание программ.	2	0	0
38	5	5.2. Реализация инкапсуляции. Представление свойств в классе. Разбор алгоритмов.	2	0	0
39	5	5.3. Организация наследуемых объектов. Представление в модуле. Представление различных областей видимости. Представление событий. Организация перекрытия статических методов. Написание программ.	2	0	0
40	5	5.3. Организация наследуемых объектов. Представление в модуле. Представление различных областей видимости. Представление событий. Организация перекрытия статических методов. Разбор алгоритмов.	2	0	0
41	5	5.4. Организация наследуемых объектов. Представление в модуле. Представление различных областей видимости. Представление событий. Организация перекрытия статических методов. Написание программ.	4	0	0
42	5	5.4. Организация наследуемых объектов. Представление в модуле. Представление различных областей видимости. Представление событий. Организация перекрытия статических методов. Разбор алгоритмов.	4	0	0

43	5	5.5. Работа с классами. Реализация методов и использование операторов для работы с классами. Работа с библиотекой классов. Написание программ.	2	0	0
44	5	5.5. Работа с классами. Реализация методов и использование операторов для работы с классами. Работа с библиотекой классов. Разбор алгоритмов.	2	0	0
45	5	5.6. Создание контейнера и методов работы с ним. Написание программ.	4	0	0
46	5	5.6. Создание контейнера и методов работы с ним. Разбор алгоритмов.	4	0	0
47	5	5.7. Использование контейнеров стандартных библиотек. Написание программ.	4	0	0
48	5	5.7. Использование контейнеров стандартных библиотек. Разбор алгоритмов.	4	0	0
49	6	6.1. Обобщенное программирование. Анонимные методы. Написание программ.	4	0	0
50	6	6.1. Обобщенное программирование. Анонимные методы. Разбор алгоритмов.	4	0	0
51	6	6.2. Визуальное программирование в среде Delphi. Создание графического интерфейса пользователя, предусматривающего использование файлов и изображений. Написание программ.	6	0	0

52	6	6.2. Визуальное программирование в среде Delphi. Создание графического интерфейса пользователя, предусматривающего использование файлов и изображений. Разбор алгоритмов.	6	0	0
53	6	6.3. Освоение базовой среды для проектирования программ с использованием языка UML. Написание программ.	2	0	0
54	6	6.3. Освоение базовой среды для проектирования программ с использованием языка UML. Разбор алгоритмов.	2	0	0
55	6	6.4. Проектирование в базовой среде простых программ с использованием диаграмм языка UML. Написание программ.	4	0	0
56	6	6.4. Проектирование в базовой среде простых программ с использованием диаграмм языка UML. Разбор алгоритмов.	4	0	0
Всего			212	0	0

### 3.4 Лабораторные занятия

№ п/п	№ раздела дисциплины	Наименование занятий	Объем в акад. часах		
			Всего	в том числе, в инновационной форме	в том числе, в электронной форме
Всего					

## 4 Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (модулю)

Авторы, составители	Заглавие	Издательство, год
---------------------	----------	-------------------

Л1.1	Кучунова Е. В., Олейников Б. В., Чередниченко О. М.	Программирование. Процедурное программирование: учебное пособие [для студентов бакалавриата по напр. 02.03.01. «Математика. Компьютерные науки»]	Красноярск: СФУ, 2016
------	--	--	--------------------------

## 5 Фонд оценочных средств для проведения промежуточной аттестации

Оценочные средства находятся в приложении к рабочим программам дисциплин.

## 6 Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины (модуля)

6.1. Основная литература			
	Авторы, составители	Заглавие	Издательство, год
Л1.1	Фаронов В. В.	TURBO PASCAL: учеб. пособие для вузов	Москва: Питер, 2009
Л1.2	Фаронов В. В.	DELPHI. Программирование на языке высокого уровня: учебник для вузов	Москва: Питер, 2009
Л1.3	Масловская О. М.	Объектно-ориентированное программирование: учебник для студентов вузов	Одесса: Укрполиграф, 2007
Л1.4	Пайлон Д., Питмен Н.	UML 2 для программистов: пер. с англ.	Санкт- Петербург: Питер, 2012
Л1.5	Вирт Н., Ткачев Ф. В.	Алгоритмы и структуры данных. Новая версия для Оберона + CD: [учебник]	Москва: ДМК Пресс, 2014
Л1.6	Фаулер М., Петухов А.	UML. Основы: краткое рук. по стандартному языку объектного моделирования	Санкт- Петербург: Символ-Плюс, 2008
Л1.7	Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К., Красиков И. В., Орехова Н. А., Романов В. Н., Красикова И. В.	Алгоритмы: построение и анализ: [учебник]	Москва: Вильямс, 2013
Л1.8	Розенберг Д., Скотт К.	Применение объектного моделирования с использованием UML и анализ прецедентов: перевод с английского	Москва: ДМК Пресс, 2007
Л1.9	Васильев А.	Объектно-ориентированное программирование: учебный курс	СПб.: Питер, 2012
Л1.10	Рамбо Д., Блах М.	UML 2.0. Объектно-ориентированное моделирование и разработка	СПб.: Питер, 2007

Л1.1 1	Кнут Д. Э.	Искусство программирования: Т. 1. Основные алгоритмы: [учебное пособие] : [перевод с английского]	Москва: Издательский дом "Вильямс", 2000
Л1.1 2	Кнут Д. Э.	Искусство программирования: Т. 3. Сортировка и поиск: [учебное пособие] : [перевод с английского]	Москва: Издательский дом "Вильямс", 2000
Л1.1 3	Кнут Д. Э.	Искусство программирования: Т. 2. Получисленные алгоритмы: [учебное пособие] : [перевод с английского]	Москва: Издательский дом "Вильямс", 2000
Л1.1 4	Иванова Г.С., Ничушкина Т.Н., Пугачев Е.К.	Объектно-ориентированное программирование: Учебник для вузов	Москва: Изд-во МГТУ им. Н.Э.Баумана, 2003
Л1.1 5	Фризен И. Г.	Основы алгоритмизации и программирования (среда PascalABC.Net): учебное пособие	Москва: Издательство "ФОРУМ", 2017
Л1.1 6	Буч Г., Рамбо Дж., Якобсон И.	Язык UML. Руководство пользователя	Москва: ДМК Пресс, 2008
Л1.1 7	Тузовский А. Ф.	Объектно-ориентированное программирование: учебное пособие для прикладного бакалавриата	Москва: Юрайт, 2017
Л1.1 8	Леоненков А. В.	Самоучитель UML: Самоучитель	Санкт-Петербург: Издательство "БХВ-Петербург", 2015
<b>6.2. Дополнительная литература</b>			
	Авторы, составители	Заглавие	Издательство, год
Л2.1	Фаронов В.В.	Турбо Паскаль 7.0. Практика программирования: учебное пособие	М.: КНОРУС, 2011
<b>6.3. Методические разработки</b>			
	Авторы, составители	Заглавие	Издательство, год
Л3.1	Кучунова Е. В., Олейников Б. В., Чередниченко О. М.	Программирование. Процедурное программирование: учебное пособие [для студентов бакалавриата по напр. 02.03.01. «Математика. Компьютерные науки»]	Красноярск: СФУ, 2016

## **7 Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины (модуля)**

Э1	Королевство Дельфи	1.	<a href="http://www.delphikingdom.com">www.delphikingdom.com</a>
Э2	Мастера DELPHI (документация, исходники)	2.	<a href="http://www.delphimaster.ru">www.delphimaster.ru</a>

Э3	AlgoList - алгоритмы, методы, исходники.	3. <a href="http://algotlist.manual.ru/">http://algotlist.manual.ru/</a>
Э4	Исходники Программирование Pascal-Паскаль - Helpov.Net	4. <a href="http://www.pascal.helpov.net/">http://www.pascal.helpov.net/</a>
Э5		<a href="https://e.sfu-kras.ru/course/view.php?id=2263">https://e.sfu-kras.ru/course/view.php?id=2263</a>
Э6		<a href="https://e.sfu-kras.ru/course/view.php?id=2261">https://e.sfu-kras.ru/course/view.php?id=2261</a>
Э7		<a href="https://e.sfu-kras.ru/course/view.php?id=2260">https://e.sfu-kras.ru/course/view.php?id=2260</a>
Э8		<a href="http://study.sfu-kras.ru/course/view.php?id=6">http://study.sfu-kras.ru/course/view.php?id=6</a>
Э9		<a href="http://study.sfu-kras.ru/course/view.php?id=121">http://study.sfu-kras.ru/course/view.php?id=121</a>

## 8 Методические указания для обучающихся по освоению дисциплины (модуля)

Дисциплина «Программирование» изучается в 1, 2 и 3 семестре.

Предусматривается три вида занятий: лекции, семинарские занятия и практические работы на компьютере.

Занятия лекционного типа проводятся один раз в две недели. Каждую неделю проводится одно семинарское занятие в лекционной аудитории и одно практическое занятие в компьютерном классе.

Курс содержит 3 модуля: 1)Процедурного программирование с использованием статической памяти - изучается в 1 семестре; 2) Сложные динамические структуры данных - изучается во 2 семестре; 3) Основы объектно-ориентированного, визуального программирования и языка UML - изучается в 3 семестре.

Каждый из модулей, в свою очередь, подразделяется на 2 раздела. Содержание всех разделов, подробно представлено в таблице 3.1.

Сроки реализации каждого раздела:

- раздел 1 "Основы программирования и разработки программ" – 1–8 неделя 1 семестра;

- раздел 2 "Создание структур данных на основе статической памяти и их обработка. Создание отдельно компилируемых подпрограмм", - 9–18 неделя 1 семестра;

- раздел 3 "Представление и использование произвольных статических структур данных. Использование динамической памяти. Организация и использование линейных динамических структур данных" - 1-8 неделя 2 семестра;

- раздел 4 "Нелинейные динамические структуры данных. Организация и использование" - 9-17 неделя 2 семестра;

- раздел 5 "Основы объектно-ориентированного

программирования" - 1–8 неделя 3 семестра;

- раздел 6 "Обобщенное, визуальное программирование и проектирование. Язык UML" - 9–18 неделя 3 семестра;

Занятия лекционного типа предназначены для изучения теоретического материала по темам разделов (представлены по-лекционно).

В рамках семинарских занятий происходит изучение и реализация основных методов, подходов и понятий, а также применение приобретенных теоретических знаний для решения практических задач обработки данных и математического моделирования (представлены по основным 8-ми темам).

В рамках практических занятий на компьютере производится выполнение работ по основным 8-ми темам. По каждой теме студенту необходимо разработать программное обеспечение

на базовом языке программирования в соответствии с заданием, которое он получает от преподавателя в начале семестра, выполнить отладку программы и её сдачу. По каждой теме предусматривается не менее 25 работ, с тем чтобы каждый студент в группе имел свое индивидуальное задание. Кроме этого, преподаватель может давать дополнительные задания отдельным студентам по их желанию в зависимости от знаний и способностей этих студентов. Выполнение этих дополнительных заданий может влиять на общую оценку по дисциплине. Сдача задания включает в себя on-line демонстрацию работы созданного студентом программного обеспечения и ответы на вопросы преподавателя по программе и изучаемой теме.

Организация процесса работы по дисциплине (модулю) «Программирование» помимо аудиторной работы предусматривает и самостоятельную работу студентов, которая в основном направлена для

- 1) самостоятельного изучения теоретического материала, необходимого для усвоения дисциплины и решения индивидуальных задач (примерно, 1/3 от общего количества часов, отводимых на самостоятельную работу студентов),

- 2) выполнения индивидуальных заданий: - 24 задания в течение 3 семестров (примерно, 2/3 от общего количества часов, отводимых на самостоятельную работу студентов).

Литература для самостоятельного изучения теоретического материала приводится в данной рабочей программе и доводится до студентов лектором.

Выдачу индивидуальных лабораторных заданий (задач) для их

самостоятельного выполнения студентами осуществляет преподаватель, ведущий практические занятия. Студент получает эти задания во время практических занятий в аудитории или может взять их с нижеуказанных сайтов. Этот же преподаватель и принимает выполненные семестровые задания по мере их готовности в течении семестра либо очно, либо заочно в режиме дистанционного взаимодействия.

Если студент зарегистрирован на сайте «Электронные курсы в СФУ в системе дистанционного обучения Moodle» (<http://e.sfu-kras.ru> или <http://study.sfu-kras.ru/>), то он имеет возможность получения всех материалов, выложенных в системе Moodle по данной дисциплине, включая и экзаменационные задания предыдущих лет (например, см., <http://study.sfu-kras.ru/course/view.php?id=6> и <http://study.sfu-kras.ru/course/view.php?id=121>),

а также имеет возможность удаленного интерактивного взаимодействия с преподавателем, который, в свою очередь, имеет инструменты on-line контроля процесса обучения студента.

Организация самостоятельной работы производится в соответствии с графиком учебного процесса и самостоятельной работы, который доводится преподавателем до студентов.

Текущий контроль практических знаний выполняется в процессе сдачи-приема лабораторных работ, выданных каждому студенту в начале семестра, а также по результатам контрольной работы, проводимой преподавателем в середине каждого семестра (по завершении изучения каждого раздела).

Промежуточными формами общего контроля являются: зачет в конце 1-го семестра, экзамен по окончании 2-го семестра, экзамен по окончании 3-го семестра.

Целью промежуточных форм контроля является проверка знаний студента и его умений применять эти знания в практике создания алгоритмов и программного обеспечения на изучаемом языке программирования.

На зачете требуется выполнить работу, включающую в себя решение практической задачи на компьютере (включающее разработку алгоритма, программирование, отладку и получение требуемого результата).

На экзамене требуется выполнить работу, включающую в себя два основных задания:

- ответ на один теоретический вопрос по материалам лекционных занятий (темы 7 – 12);
- решение практической задачи на компьютере (включающее

разработку алгоритма, программирование, отладку и получение требуемого результата).

При подведении общих итогов обучения обязательно учитываются результаты текущего контроля: выполнения контрольных работ и выполнения лабораторных работ на компьютере.

Организация самостоятельной работы производится в соответствии с графиком учебного процесса и самостоятельной работы, который доводится преподавателем до студентов.

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

Для лиц с нарушениями зрения:

– в форме электронного документа.

Для лиц с нарушениями слуха:

– в печатной форме,

– в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

– в печатной форме,

– в форме электронного документа

## **9 Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю) (при необходимости)**

### **9.1 Перечень необходимого программного обеспечения**

9.1.1	MS Power Point, MS Word или аналогичное
9.1.2	Embarcadero Delphi 2009 или более поздних версий (желательно Delphi XE10). Возможно (особенно в рамках самостоятельной работы студентов в первых двух семестрах) использование PascalABC.Net
9.1.3	Визуализаторы рекурсивных алгоритмов, структур данных, ПО для построения фракталов. Это ПО можно найти в открытом доступе сети Интернет.
9.1.4	Свободно распространяемое ПО для построения UML-диаграмм, например, Star UML

## 9.2 Перечень необходимых информационных справочных систем

9.2.1	Отдельно не требуются. Как правило, они сопровождают используемое ПО
-------	--

### **10 Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине (модулю)**

Материально-техническое обеспечение образовательного процесса должно включать

- 1) Компьютерные классы, с рабочим местом преподавателя, подключенные к сети Интернет
- 2) Средства интерактивного отображения и взаимодействия (видеопроектор, интерактивная доска)
- 3) Wi-Fi сеть для организации взаимодействия с мобильными устройствами обучающихся
- 4) Желательна организация сетевого интерактивного взаимодействия, например, таких технологий, как Smart class
- 5) Кроме этого каждый студент должен иметь средства индивидуального хранения информации (flash-карта, portable disk и т.п.)
- 6) Для выполнения самостоятельных работ (в том числе и дома) желательно, чтобы каждый студент имел собственный индивидуальный компьютер.

Освоение дисциплины инвалидами и лицами с ограниченными возможностями здоровья, в зависимости от нозологий, осуществляется с использованием средств обучения общего и специального назначения.