

Министерство науки и высшего образования РФ
Федеральное государственное автономное образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

Б1.О.26 Программирование

наименование дисциплины (модуля) в соответствии с учебным планом

Направление подготовки / специальность

02.03.01 Математика и компьютерные науки

Направленность (профиль)

02.03.01.31 Математическое и компьютерное моделирование

Форма обучения

очная

Год набора

2023

Красноярск 2023

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

Программу составили _____

к.ф.н., доцент, Олейников Б.В.

должность, инициалы, фамилия

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

Код и наименование индикатора достижения компетенции	Запланированные результаты обучения по дисциплине
ОПК-4: Способен находить, анализировать, реализовывать программно и использовать на практике математические алгоритмы, в том числе с применением современных вычислительных систем	
ОПК-4.1: Находит и анализирует математические алгоритмы для решения практических задач	математические алгоритмы для решения практических задач, способы их сравнения анализировать математические алгоритмы по эффективности, быстродействию, использованию памяти методами тестирования математических алгоритмов
ОПК-4.2: Реализовывает программно и использует на практике математические алгоритмы, в том числе с применением современных вычислительных систем	математические алгоритмы, структуры данных Реализовывать программно математические алгоритмы, в том числе с применением современных вычислительных систем языком программирования высокого уровня, средами разработки и отладки программ
ОПК-5: Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной	

деятельности	
ОПК-5.1: Использует электронные библиотечные системы, национальные и международные базы данных для поиска необходимой научной литературы	Электронные библиотечные системы, национальные базы данных для поиска информации, касающейся математики и программирования Осуществлять поиск информации, необходимой для решения задач в области профессиональной деятельности стратегиями эффективного поиска информации в электронных библиотечных системах, национальных и международных базах данных
ОПК-5.2: Применяет современное программное обеспечение для решения различных задач профессиональной деятельности	классификацию и назначение современного программного обеспечения, методы эффективной работы с ним. использовать современное программное обеспечение для решения профессиональных задач программным обеспечением для разработки программ с целью решения профессиональных задач.
ОПК-6: Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения	
ОПК-6.1: Разрабатывает алгоритмы решения прикладных задач с использованием математических и аналитических методов	математические и аналитические методы решения прикладных задач, способы разработки эффективных алгоритмов применять на практике математические и аналитические методы для разработки эффективных алгоритмов математическими и аналитическими методами разработки алгоритмов решения прикладных задач
ОПК-6.2: Реализует алгоритмы с использованием современных средств разработки прикладного программного обеспечения	современные средства разработки разработки программного обеспечения, методы разработки алгоритмов использовать современные средства разработки программного обеспечения для разработки программ навыками разработки алгоритмов на языке высокого уровня с использованием современных средств разработки программного обеспечения
ОПК-6.3: Применяет на практике знания основных положений и концепций прикладного и системного программирования, архитектуры компьютеров	основные положения и концепции прикладного и системного программного обеспечения, архитектуры компьютеров. применять на практике знания основные положения и концепции прикладного и системного программного обеспечения, архитектуры компьютеров. методами разработки прикладного программного обеспечения

1.4 Особенности реализации дисциплины

Язык реализации дисциплины: Русский.

Дисциплина (модуль) реализуется с применением ЭО и ДОТ

URL-адрес и название электронного обучающего курса: <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
Контактная работа с преподавателем:	7,39 (266)			
занятия лекционного типа	1,5 (54)			
практические занятия	5,89 (212)			
Самостоятельная работа обучающихся:	3,61 (130)			
курсовое проектирование (КП)	Нет			
курсовая работа (КР)	Нет			
Промежуточная аттестация (Зачёт) (Экзамен)	2 (72)			

3 Содержание дисциплины (модуля)

3.1 Разделы дисциплины и виды занятий (тематический план занятий)

№ п/п		Модули, темы (разделы) дисциплины		Контактная работа, ак. час.								
				Занятия лекционного типа		Занятия семинарского типа				Самостоятельная работа, ак. час.		
						Семинары и/или Практические занятия		Лабораторные работы и/или Практикумы				
						Всего	В том числе в ЭИОС	Всего	В том числе в ЭИОС			Всего
1. 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								
		2. 1.1. Структура программы. Операторы ввода вывода. Числовые типы данных. Логические и арифметические выражения. Блочный оператор. Условный оператор. (Написание программ)				2						
		3. Структура программы. Операторы ввода вывода. Числовые типы данных. Логические и арифметические выражения. Блочный оператор. Условный оператор. (Разбор алгоритмов)				2						
		4. 1.2. Типизация данных. Основные типы данных и средства работы с ними. Основные операторы		2								

5. 1.2. Основные типы простых данных. Основные операции. Операторы цикла. Операторы выбора. (Написание программ)			4					
6. 1.2. Основные типы простых данных. Основные операции. Операторы цикла. Операторы выбора. (Разбор алгоритмов)			4					
7. 1.3. Проектируемые типы для числовых данных. Основные алгоритмы для работы с ними	2							
8. 1.3. Проектируемые типы для числовых данных. Массивы. Основные алгоритмы для работы с ними. Сортировки. (Написание программ)			6					
9. 1.3. Проектируемые типы для числовых данных. Массивы. Основные алгоритмы для работы с ними. Сортировки. (Разбор алгоритмов)			6					
10. 1.4. Основы и инструменты структурного программирования. Процедурные типы. Рекуррентность и рекурсия. Реализация в языке	2							
11. 1.4. Организация подпрограмм. Процедуры и функции. Процедурные типы. Прямая и косвенная рекурсия. Построение рекурсивных подпрограмм			4					
12. 1.4. Организация подпрограмм. Процедуры и функции. Процедурные типы. Прямая и косвенная рекурсия. Построение рекурсивных подпрограмм			4					
13. 1.5. Мат. индукция, рекурсия и итерация. Фракталы	2							
14. 1.5. Построение геометрических фракталов. Примеры использования генераторов фракталов. (Написание программ)			2					

15. 1.5. Построение геометрических фракталов. Примеры использования генераторов фракталов (Разбор алгоритмов)			2					
16. Самостоятельная работа по разделу 1.							18	
2. 2								
1. 2.1. Нечисловые типы данных (строки, множества) и инструменты работы с ними	2							
2. 2.1. Нечисловые типы данных (строки, множества) и инструменты работы с ними. (Написание программ)			4					
3. 2.1. Нечисловые типы данных (строки, множества) и инструменты работы с ними. (Разбор алгоритмов)			4					
4. 2.2. Комбинированный тип данных. Файловый тип данных. Основные инструменты по работе с ними	2							
5. 2.2. Комбинированный тип данных. Файловый тип данных. Основные инструменты по работе с ними. Написание программ.			6					
6. 2.2. Комбинированный тип данных. Файловый тип данных. Основные инструменты по работе с ними. Разбор алгоритмов.			6					
7. 2.3. Расширенная работа с файлами. Типизированные константы. Преобразование типов	2							
8. 2.3. Расширенная работа с файлами. Типизированные константы. Преобразование типов. (Написание программ)			6					
9. 2.3. Расширенная работа с файлами. Типизированные константы. Преобразование типов. (Разбор алгоритмов)			6					
10. 2.4. Обработка исключительных ситуаций. Создание отдельно компилируемых подпрограмм.	2							

11. 2.4. Обработка исключительных ситуаций. Создание отдельно компилируемых подпрограмм. (Написание программ)			2					
12. 2.4. Обработка исключительных ситуаций. Создание отдельно компилируемых подпрограмм. (Разбор алгоритмов)			2					
13. Самостоятельная работа по разделу 2.							18	
3.3								
1. 3.1. Уровни представления данных. Структуры данных. Представление структур. Графы	2							
2. 3.1. Представление в программе графовых структур. Решение задач (основное дерево, кратчайший путь). Написание программ.			2					
3. 3.1. Представление в программе графовых структур. Решение задач (основное дерево, кратчайший путь). Разбор алгоритмов			2					
4. 3.2. Статическая и динамическая память. Адресация памяти в архитектуре Intel 8086/88. Средства работы с динамической памятью в базовом языке	2							
5. 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					
6. 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					

7. 3.3. Линейные динамические структуры данных. Работа с ними. Области применения	2							
8. 3.3. Организация линейных динамических структур данных (стек, очередь, дек, кольцо). Работа с ними. Написание программ.			6					
9. 3.3. Организация линейных динамических структур данных (стек, очередь, дек, кольцо). Работа с ними. Разбор алгоритмов.			6					
10. 3.4. Большие числа. Их представление линейными динамическими структурами. Реализация основных операций над ними	2							
11. 3.4. Представление больших чисел линейными динамическими структурами. Реализация основных арифметических операций на них. Написание программ.			4					
12. 3.4. Представление больших чисел линейными динамическими структурами. Реализация основных арифметических операций на них. Разбор алгоритмов.			4					
13. Самостоятельная работа по разделу 3.							20	
4. 4								
1. 4.1. Виды нелинейных динамических структур. Бинарные деревья. Операции над ними. Прошитые деревья. Области применения	2							
2. 4.1. Построение бинарных деревьев и реализация основных операций над ними. Создание прошитых деревьев. Написание программ.			6					
3. 4.1. Построение бинарных деревьев и реализация основных операций над ними. Создание прошитых деревьев. Разбор алгоритмов.			6					

4. 4.2. Поиск на деревьях. Сбалансированные деревья. Построение и использование AVL-деревьев	2							
5. 4.2. Балансировка деревьев вращением. Построение и использование AVL-деревьев. Написание программ.			4					
6. 4.2. Балансировка деревьев вращением. Построение и использование AVL-деревьев. Разбор алгоритмов.			4					
7. 4.3. «Цветная» балансировка деревьев. RB- и AA-деревья. Построение и использование	2							
8. 4.3. «Цветная» балансировка деревьев. Построение и использование RB- и AA-деревьев. Написание программ.			4					
9. 4.3. «Цветная» балансировка деревьев. Построение и использование RB- и AA-деревьев. Разбор алгоритмов.			4					
10. 4.4. N-арные деревья. Способы представления. B-деревья. Основные операции. Крупномасштабные N-арные деревья – как основа представления данных в СУБД.	4							
11. 4.4. Создание N-арных деревьев. Создание B-деревьев. Реализация основных операций. Написание программ.			4					
12. 4.4. Создание N-арных деревьев. Создание B-деревьев. Реализация основных операций. Разбор алгоритмов.			4					
13. Самостоятельная работа по разделу 4.							20	
5. 5								
1. 5.1. Введение. Модели мира и парадигмы программирования. ООМ. История. Среды. Языки	2							

2. 5.1. Анатомия класса и объекта. Примеры построения. Написание программ.			2					
3. 5.1. Анатомия класса и объекта. Примеры построения. Разбор алгоритмов.			2					
4. 5.2.ООП. Основные понятия. Концепция. Абстракция. Инкапсуляция. Поля, методы. Конструктор, деструктор. Свойства. Классификация свойств	2							
5. 5.2. Реализация инкапсуляции. Представление свойств в классе. Написание программ.			2					
6. 5.2. Реализация инкапсуляции. Представление свойств в классе. Разбор алгоритмов.			2					
7. 5.3.Наследование. Модуль. Области видимости. События. Перекрытие методов. Виды методов. Полиморфизм. Класс и объект «изнутри».	2							
8. 5.3. Организация наследуемых объектов. Представление в модуле. Представление различных областей видимости. Представление событий. Организация перекрытия статических методов. Написание программ.			2					
9. 5.3. Организация наследуемых объектов. Представление в модуле. Представление различных областей видимости. Представление событий. Организация перекрытия статических методов. Разбор алгоритмов.			2					
10. 5.4. Обработка сообщений. Перегрузка методов. Событие. Делегирование событий. Работа с классами. Методы и операторы для работы с классами. Библиотека классов VCL.	2							

11. 5.4. Организация наследуемых объектов. Представление в модуле. Представление различных областей видимости. Представление событий. Организация перекрытия статических методов. Написание программ.			4					
12. 5.4. Организация наследуемых объектов. Представление в модуле. Представление различных областей видимости. Представление событий. Организация перекрытия статических методов. Разбор алгоритмов.			4					
13. 5.5. Контейнер. Итератор. Интерфейсы. Стандартные контейнеры в библиотеке VCL и других.	2							
14. 5.5. Работа с классами. Реализация методов и использование операторов для работы с классами. Работа с библиотекой классов. Написание программ.			2					
15. 5.5. Работа с классами. Реализация методов и использование операторов для работы с классами. Работа с библиотекой классов. Разбор алгоритмов.			2					
16. 5.6. Создание контейнера и методов работы с ним. Написание программ.			4					
17. 5.6. Создание контейнера и методов работы с ним. Разбор алгоритмов.			4					
18. 5.7. Использование контейнеров стандартных библиотек. Написание программ.			4					
19. 5.7. Использование контейнеров стандартных библиотек. Разбор алгоритмов.			4					
20. Самостоятельная работа по разделу 5.							20	
6. 6								

1. 6.1. Обобщенное программирование. Анонимные методы	2							
2. 6.1. Обобщенное программирование. Анонимные методы. Написание программ.			4					
3. 6.1. Обобщенное программирование. Анонимные методы. Разбор алгоритмов.			4					
4. 6.2. Визуальное программирование в среде Delphi. Создание графического интерфейса пользователя. Использование файлов и изображений	2							
5. 6.2. Визуальное программирование в среде Delphi. Создание графического интерфейса пользователя, предусматривающего использование файлов и изображений. Написание программ.			6					
6. 6.2. Визуальное программирование в среде Delphi. Создание графического интерфейса пользователя, предусматривающего использование файлов и изображений. Разбор алгоритмов.			6					
7. 6.3. Жизненный цикл ПО. ООМ. Язык UML – как инструмент ООМ. История, понятия, структура языка UML	2							
8. 6.3. Освоение базовой среды для проектирования программ с использованием языка UML. Написание программ.			2					
9. 6.3. Освоение базовой среды для проектирования программ с использованием языка UML. Разбор алгоритмов.			2					
10. 6.4. Основные диаграммы языка UML. Назначение. Нотация. Примеры. Среда поддержки языка UML	2							

11. 6.4. Проектирование в базовой среде простых программ с использование диаграмм языка UML. Написание программ.			4					
12. 6.4. Проектирование в базовой среде простых программ с использование диаграмм языка UML. Разбор алгоритмов.			4					
13. Самостоятельная работа по разделу 6.							34	
Всего	54		212				130	

4 Учебно-методическое обеспечение дисциплины

4.1 Печатные и электронные издания:

1. Фаронов В. В. TURBO PASCAL: учеб. пособие для вузов(Москва: Питер).
2. Фаронов В. В. DELPHI. Программирование на языке высокого уровня: учебник для вузов(Москва: Питер).
3. Масловская О. М. Объектно-ориентированное программирование: учебник для студентов вузов(Одесса: Укрполиграф).
4. Пайлон Д., Питмен Н. UML 2 для программистов: пер. с англ.(Санкт-Петербург: Питер).
5. Вирт Н., Ткачев Ф. В. Алгоритмы и структуры данных. Новая версия для Oberon + CD: [учебник](Москва: ДМК Пресс).
6. Фаулер М., Петухов А. UML. Основы: краткое рук. по стандартному языку объектного моделирования(Санкт-Петербург: Символ-Плюс).
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).
8. Розенберг Д., Скотт К. Применение объектного моделирования с использованием UML и анализ прецедентов: перевод с английского (Москва: ДМК Пресс).
9. Васильев А. Объектно-ориентированное программирование: учебный курс(СПб.: Питер).
10. Рамбо Д., Блаха М. UML 2.0. Объектно-ориентированное моделирование и разработка(СПб.: Питер).
11. Кнут Д. Э. Искусство программирования: Т. 1. Основные алгоритмы: [учебное пособие] : [перевод с английского](Москва: Издательский дом "Вильямс").
12. Кнут Д. Э. Искусство программирования: Т. 3. Сортировка и поиск: [учебное пособие] : [перевод с английского](Москва: Издательский дом "Вильямс").
13. Кнут Д. Э. Искусство программирования: Т. 2. Получисленные алгоритмы: [учебное пособие] : [перевод с английского](Москва: Издательский дом "Вильямс").
14. Иванова Г.С., Ничушкина Т.Н., Пугачев Е.К. Объектно-ориентированное программирование: Учебник для вузов(Москва: Изд-во МГТУ им. Н.Э.Баумана).
15. Фризен И. Г. Основы алгоритмизации и программирования (среда PascalABC.Net): учебное пособие(Москва: Издательство "ФОРУМ").
16. Буч Г., Рамбо Дж., Якобсон И. Язык UML. Руководство пользователя (Москва: ДМК Пресс).
17. Тузовский А. Ф. Объектно-ориентированное программирование: учебное пособие для прикладного бакалавриата(Москва: Юрайт).
18. Леоненков А. В. Самоучитель UML: Самоучитель(Санкт-Петербург: Издательство "БХВ-Петербург").

19. Фаронов В.В. Турбо Паскаль 7.0. Практика программирования: учебное пособие(М.: КНОРУС).
20. 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).

4.2 Лицензионное и свободно распространяемое программное обеспечение, в том числе отечественного производства (программное обеспечение, на которое университет имеет лицензию, а также свободно распространяемое программное обеспечение):

1. MS Power Point, MS Word или аналогичное
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
3. Визуализаторы рекурсивных алгоритмов, структур данных, ПО для построения фракталов. Это ПО можно найти в открытом доступе сети Интернет.
4. Свободно распространяемое ПО для построения UML-диаграмм, например, Star UML

4.3 Интернет-ресурсы, включая профессиональные базы данных и информационные справочные системы:

1. Отдельно не требуются. Как правило, они сопровождают используемое ПО

5 Фонд оценочных средств

Оценочные средства находятся в приложении к рабочим программам дисциплин.

6 Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине (модулю)

Материально-техническое обеспечение образовательного процесса должно включать

Компьютерные классы, с рабочим местом преподавателя, подключенные к сети Интернет

Средства интерактивного отображения и взаимодействия (видеопроектор, интерактивная доска)

Wi-Fi сеть для организации взаимодействия с мобильными устройствами обучающихся

Желательна организация сетевого интерактивного взаимодействия, например, таких технологий, как Smart class

Кроме этого каждый студент должен иметь средства индивидуального хранения информации (flash-карта, portable disk и т.п.)

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.