

АВТОНОМНАЯ НЕКОММЕРЧЕСКАЯ ОРГАНИЗАЦИЯ
ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ СОЦИАЛЬНЫЙ ИНСТИТУТ»

Утверждаю
Декан факультета
_____ Ж.В. Игнатенко
«28» _____ 10 _____ 2020г.



РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Технологии программирования

Направление подготовки: 09.03.02 Информационные системы и технологии

Направленность (профиль) программы: Информационные технологии в управлении предприятием

Квалификация выпускника: бакалавр

Форма обучения: очная, заочная

Год начала подготовки – 2019

Разработана
Канд. эконом. наук, доцент, доцент
_____ А.Ю. Орлова

Согласована
зав. выпускающей кафедры
_____ А.Ю. Орлова

Рекомендована
на заседании кафедры
от «28» 10 _____ 2020г.
протокол № 2
Зав. кафедрой
_____ Ж.В. Игнатенко

Одобрена
на заседании учебно-методической
комиссии факультета
от «28» 10 _____ 2020г.
протокол № 2
Председатель УМК
_____ Ж.В. Игнатенко

Ставрополь, 2020 г.

Содержание

1. Цели освоения дисциплины	3
2. Место дисциплины в структуре ОПОП.....	3
3. Планируемые результаты обучения по дисциплине.....	4
4. Объем дисциплины и виды учебной работы	5
5. Содержание и структура дисциплины.....	6
5.1. Содержание дисциплины.....	6
5.2. Структура дисциплины	7
5.3. Занятия семинарского типа	8
5.4. Курсовой проект (курсовая работа, расчетно-графическая работа, реферат, контрольная работа).....	9
5.5. Самостоятельная работа	9
6. Образовательные технологии.....	10
7. Фонд оценочных средств (оценочные материалы) для текущего контроля успеваемости, промежуточной аттестации	11
8. Учебно-методическое и информационное обеспечение дисциплины	11
8.1. Основная литература.....	11
8.2. Дополнительная литература.....	11
8.3 Программное обеспечение.....	12
8.4 Профессиональные базы данных	12
8.5. Информационные справочные системы.....	12
8.6. Интернет-ресурсы.....	12
8.7. Методические указания по освоению дисциплины.....	12
9. Материально-техническое обеспечение дисциплины	14
10. Особенности освоения дисциплины лицами с ограниченными возможностями здоровья.....	15
Приложение 1.....	Ошибка! Закладка не определена.

1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Целями освоения дисциплины «Технологии программирования» являются: формирование навыков по разработке, документированию и сопровождению программного обеспечения; реализация требований, установленных в квалификационной характеристике, при подготовке бакалавров. Изучение дисциплины опирается на знания и навыки программирования в средах с процедурной парадигмой, знания из математических курсов.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

Дисциплина (Б.1.Б.19) «Технологии программирования» входит в обязательную часть Блока 1 «Дисциплины (модули)» и находится в логической и содержательно-методической связи с другими дисциплинами.

Предшествующие дисциплины (курсы, модули, практики)	Последующие дисциплины (курсы, модули, практики)
Информатика и программирование	Визуальное программирование
Языки программирования	Интернет-программирование

Требования к «входным» знаниям, умениям и навыкам обучающегося, необходимым при освоении данной дисциплины

Знать:

- назначение и виды информационных моделей, описывающих реальные объекты или процессы;
- использование алгоритма как способа автоматизации деятельности;
- назначение и функции операционных систем;
- принципы построения архитектуры программного обеспечения и виды архитектуры программного обеспечения, типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке программного обеспечения, методы и средства проектирования программного обеспечения, баз данных и программных интерфейсов;
- правила настройки и эксплуатации устанавливаемого системного программного обеспечения, включая лицензионные требования;
- основы архитектуры, устройства и функционирования вычислительных систем;
- принципы организации, состав и схемы работы операционных систем; стандарты информационного взаимодействия систем;
- требования охраны труда при работе с аппаратными, программно- аппаратными и программными средствами администрируемой инфокоммуникационной системы;
- основы обеспечения информационной безопасности; нормативные правовые акты в области информационной безопасности;
- инструкции по настройке и эксплуатации устанавливаемого системного программного обеспечения, включая знания о типовых уязвимостях;
- регламенты обеспечения информационной безопасности;
- языки программирования и работы с базами данных, инструменты и методы проектирования и дизайна ИС, верификации структуры программного кода, инструменты и методы проектирования и верификации структуры баз данных.

Уметь:

- создавать информационные объекты сложной структуры, в том числе гипертекстовые;
- просматривать, создавать, редактировать, сохранять записи в базах данных;
- осуществлять поиск информации в базах данных, компьютерных сетях и пр.;
- представлять числовую информацию различными способами (таблица, массив, график, диаграмма и пр.);

- кодировать на языках программирования, верифицировать структуру программного кода, разрабатывать и верифицировать структуру баз данных;
- идентифицировать инциденты при работе системного программного обеспечения; применять специализированные программно-аппаратные средства для локализации инцидентов при работе системного программного обеспечения;
- выполнять настройку системного программного обеспечения в соответствии с регламентами обеспечения информационной безопасности; проводить авторизацию пользователей, имеющих доступ к настройке системного программного обеспечения инфокоммуникационной системы организации;
- применять программные и программно-аппаратные средства защиты информации;
- использовать существующие типовые решения и шаблоны проектирования программного обеспечения, применять методы и средства проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов.

Владеть навыками:

- стандартными приемами написания на алгоритмическом языке программы для решения стандартной задачи с использованием основных конструкций программирования и отладки таких программ;
- компьютерными средствами представления и анализа данных;
- проведения анализа исполнения требований к ПО, выработки вариантов реализации требований к ПО, оценки и обоснования рекомендуемых решений по ПО, выбора средств и вариантов реализации требований к программному обеспечению;
- разработки, изменения и согласования архитектуры программного обеспечения с системным аналитиком и архитектором программного обеспечения, проектирования структуры данных, базы данных и программного интерфейса, оценивания и согласования сроков выполнения задач проектирования;
- разработки и верификации структуры программного кода ИС относительно архитектуры ИС и требований заказчика к ИС, разработки структуры баз данных ИС в соответствии с архитектурной спецификацией, верификации структуры баз данных ИС относительно архитектуры ИС и требований заказчика к ИС, устранения обнаруженных несоответствий ИС требованиям.

3. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Процесс изучения дисциплины направлен на формирование элементов следующих компетенций по данному направлению (специальности):

Способен исследовать современные проблемы и методы прикладной информатики и развития информационного общества (ОПК-6).

Формируемые компетенции (код компетенции, уровень)	Планируемые результаты обучения на данном этапе формирования компетенции
ОПК- 6 Способен разрабатывать алгоритмы и программы, пригодные для практического применения в области информационных систем и технологий	Знать: содержание, объекты и субъекты информационного общества, критерии эффективности его функционирования; структуру интеллектуального капитала, проблемы инвестиций в экономику информатизации и методы оценки эффективности; правовые, экономические, социальные и психологические аспекты информатизации; теоретические проблемы прикладной информатики, в том числе семантической обработки информации, развитие представлений об оценке качества информации в

	информационных системах; современные методы, средства, стандарты информатики для решения прикладных задач различных классов; правовые, экономические, социальные и психологические аспекты информатизации деятельности организационно-экономических систем.
	Уметь: проводить анализ современных методов и средств информатики для решения прикладных задач различных классов.
	Владеть: навыками применения современных программно-технических средств для решения прикладных задач различных классов.

4. ОБЪЕМ ДИСЦИПЛИНЫ И ВИДЫ УЧЕБНОЙ РАБОТЫ

Общая трудоемкость дисциплины составляет 5 зачетных единиц, 180 академических часов.

Очная форма обучения

Вид учебной работы	Всего часов	Триместры
		7
	ОФО	ОФО
Контактная работа (всего)	50	50
в том числе:		
1) занятия лекционного типа (ЛК)	20	20
из них		
-лекций	20	20
2) занятия семинарского типа (ПЗ)	30	30
-семинары (С)		
-практические занятия (ПР)	30	30
-лабораторные работы (ЛР)		
3) групповые консультации		
4) индивидуальная работа		
5) промежуточная аттестация		
Самостоятельная работа (всего) (СР)	130	130
в том числе:		
Курсовой работа		
Расчетно-графические работы		
Контрольная работа		
Реферат		
Самоподготовка (самостоятельное изучение разделов, проработка и повторение лекционного материала и материала учебников и учебных пособий, подготовка к лабораторным и практическим занятиям, коллоквиумам, рубежному контролю и т.д.)	130	130
Общий объем, час	180	180
Форма промежуточной аттестации		диф. зачет

Заочная форма обучения

Вид учебной работы	Всего часов	Триместры
		7
	ЗФО	ЗФО
Контактная работа (всего)	18,3	18,3
в том числе:		
1) занятия лекционного типа (ЛК)	6	6
из них		
-лекций	6	6
2) занятия семинарского типа (ПЗ)	12	12
-семинары (С)		
-практические занятия (ПР)	12	12
-лабораторные работы (ЛР)		
3) групповые консультации		
4) индивидуальная работа		
5) промежуточная аттестация	0,3	0,3
Самостоятельная работа (всего) (СР)	161,7	161,7
в том числе:		
Курсовая работа		
Расчетно-графические работы		
Контрольная работа		
Реферат		
Самоподготовка (самостоятельное изучение разделов, проработка и повторение лекционного материала и материала учебников и учебных пособий, подготовка к лабораторным и практическим занятиям, коллоквиумам, рубежному контролю и т.д.)	158	158
Подготовка к аттестации	3,7	3,7
Общий объем, час	180	180
Форма промежуточной аттестации		диф. зачет

5. СОДЕРЖАНИЕ И СТРУКТУРА ДИСЦИПЛИНЫ

5.1. Содержание дисциплины

№ раздела (темы)	Наименование раздела (темы)	Содержание раздела (темы)
1.	Тема 1. Надежное программное средство как продукт технологии программирования. Исторический и социальный	Программа как формализованное описание процесса обработки данных. Программное средство. Неконструктивность понятия правильной программы. Надежность программного средства. Технология программирования как технология разработки
2.	Тема 2. Источники ошибок в программных средствах	Интеллектуальные возможности человека. Неправильный перевод как причина ошибок в программных средствах. Модель перевода. Основные

3.	Тема 3. Общие принципы разработки программных средств	Специфика разработки программных средств. Жизненный цикл программного средства. Понятие качества программного средства. Обеспечение надежности - основной мотив разработки программных средств. Методы борьбы со сложностью. Обеспечение точности перевода.
4.	Тема 4. Внешнее описание программного средства	Назначение внешнего описания программного средства и его роль в обеспечении качества программного средства. Определение требований к программному средству. Спецификация качества программного средства. Функциональная спецификация программного средства. Методы
5.	Тема 5. Методы спецификации семантики функций	Основные подходы к спецификации семантики функций. Метод таблиц решений. Операционная семантика. Денотационная семантика.
6.	Тема 6. Архитектура программного средства	Понятие архитектуры программного средства. Основные классы архитектур программных средств. Архитектурные функции. Контроль архитектуры
7.	Тема 7. Разработка структуры программы и модульное	Цель модульного программирования. Основные характеристики программного модуля. Методы разработки структуры программы. Контроль
8.	Тема 8. Разработка программного модуля	Порядок разработки программного модуля. Структурное программирование. Пошаговая детализация и понятие о псевдокоде. Контроль
9.	Тема 9. Доказательство свойств программ	Обоснования программ. Формализация свойств программ. Свойства простых операторов.
10.	Тема 10. Тестирование и отладка программного средства	Основные понятия. Принципы и виды отладки программного средства. Заповеди отладки программного средства. Автономная отладка
11.	Тема 11. Обеспечение функциональности и надежности программного средства	Функциональность и надежность как обязательные критерии качества программного средства. Обеспечение завершенности программного средства. Обеспечение точности программного средства. Обеспечение автономности программного средства.
12.	Тема 12. Обеспечение качества программного средства	Общая характеристика процесса обеспечения качества программного средства. Обеспечение легкости применения программного средства. Обеспечение эффективности программного средства. Обеспечение сопровождаемости программного средства.
13.	Тема 13. Документирование программных средств	Документация, создаваемая и используемая в процессе разработки программных средств. Пользовательская документация программных средств. Документация по сопровождению программных средств.

5.2. Структура дисциплины

Очная форма обучения

№ раздела (темы)	Наименование раздела (темы)	Количество часов			
		Всего	Л	ПЗ (ПР)	СР
1.	Надежное программное средство как продукт технологии программирования. Исторический и	16	2	2	10

	социальный контекст программирования				
2.	Источники ошибок в программных средствах	16	2	4	10
3.	Общие принципы разработки программных средств	14	2	4	10
4.	Внешнее описание программного средства	14	2	4	10
5.	Методы спецификации семантики функций	14	2	-	10
6.	Архитектура программного средства	14	2	2	10
7.	Разработка структуры программы и модульное программирование	14	2	4	10
8.	Разработка программного модуля	14	2	-	10
9.	Доказательство свойств программ	14	2	2	10
10.	Тестирование и отладка программного средства	14	2	4	10
11.	Обеспечение функциональности и надежности программного средства	12	-	-	10
12.	Обеспечение качества программного средства	12	-	-	10
13.	Документирование программных средств	12	-	4	10
	Общий объем:	180	20	30	130

Заочная форма обучения

№ раздела (темы)	Наименование раздела (темы)	Количество часов			
		Всего	Л	ПЗ (ПР)	СР
1.	Надежное программное средство как продукт технологии программирования. Исторический и социальный контекст программирования	14	2	2	12
2.	Источники ошибок в программных средствах	16	2	2	12
3.	Общие принципы разработки программных средств	14	-	-	12
4.	Внешнее описание программного средства	14	-	2	12
5.	Методы спецификации семантики функций	14	-	-	12
6.	Архитектура программного средства	14	-	-	12
7.	Разработка структуры программы и модульное программирование	16	2	2	12
8.	Разработка программного модуля	12	-	-	12
9.	Доказательство свойств программ	12	-	-	12
10.	Тестирование и отладка программного средства	12	-	2	12
11.	Обеспечение функциональности и надежности программного средства	12	-	-	12
12.	Обеспечение качества программного средства	12	-	-	12
13.	Документирование программных средств	14	-	2	14
	Диф. зачет	4	-	-	4
	Общий объем:	180	6	12	162

5.3. Занятия семинарского типа

Очная форма обучения

№ п/п	№ раздела (темы)	Вид занятия	Наименование	Количество часов
				ОФО
1	1	ПР	Динамическая память в языке C++	2
2	2	ПР	Линейный список как динамическая	4

			структура данных	
3	3	ПР	Двухсвязные списки	4
4	4	ПР	Стек как динамическая структура данных	4
5	6	ПР	Очередь как динамическая структура данных	2
6	7	ПР	Двусторонняя очередь как динамическая структура данных	4
7	9	ПР	Графы как динамическая структура данных	2
8	10	ПР	Деревья как динамическая структура данных	4
9	13	ПР	Манипуляторы в языке C++	4

Заочная форма обучения

№ п/п	№ раздела (темы)	Вид занятия	Наименование	Количество часов
				ОФО
1	1	ПР	Динамическая память в языке C++	2
2	2	ПР	Линейный список как динамическая структура данных	2
3	4	ПР	Стек как динамическая структура данных	2
4	7	ПР	Двусторонняя очередь как динамическая структура данных	2
5	10	ПР	Деревья как динамическая структура данных	2
6	13	ПР	Манипуляторы в языке C++	2

5.4. Курсовой проект (курсовая работа, расчетно-графическая работа, реферат, контрольная работа)

Не предусмотрены

5.5. Самостоятельная работа

Очная форма обучения

№ раздела (темы)	Виды самостоятельной работы	Количество часов
		ОФО
1	Проработка и повторение лекционного материала. Подготовка к лабораторной работе	10
2	Проработка и повторение лекционного материала. Подготовка к лабораторной работе	10
3	Проработка и повторение лекционного материала. Подготовка к лабораторной работе	10
4	Проработка и повторение лекционного материала. Подготовка к лабораторной работе	10
5	Проработка и повторение лекционного материала. Подготовка к лабораторной работе	10
6	Проработка и повторение лекционного материала. Подготовка к лабораторной работе	10
7	Проработка и повторение лекционного материала. Подготовка к лабораторной работе	10

8	Проработка и повторение лекционного материала. Подготовка к лабораторной работе	10
9	Проработка и повторение лекционного материала. Подготовка к лабораторной работе	10
10	Проработка и повторение лекционного материала. Подготовка к лабораторной работе	10
11	Проработка и повторение лекционного материала. Подготовка к лабораторной работе	10
12	Проработка и повторение лекционного материала. Подготовка к лабораторной работе	10
13	Проработка и повторение лекционного материала. Подготовка к лабораторной работе	10

Заочная форма обучения

№ раздела (темы)	Виды самостоятельной работы	Количество часов
		ОФО
1	Проработка и повторение лекционного материала. Подготовка к лабораторной работе	12
2	Проработка и повторение лекционного материала. Подготовка к лабораторной работе	12
3	Проработка и повторение лекционного материала. Подготовка к лабораторной работе	12
4	Проработка и повторение лекционного материала. Подготовка к лабораторной работе	12
5	Проработка и повторение лекционного материала. Подготовка к лабораторной работе	12
6	Проработка и повторение лекционного материала. Подготовка к лабораторной работе	12
7	Проработка и повторение лекционного материала. Подготовка к лабораторной работе	12
8	Проработка и повторение лекционного материала. Подготовка к лабораторной работе	12
9	Проработка и повторение лекционного материала. Подготовка к лабораторной работе	12
10	Проработка и повторение лекционного материала. Подготовка к лабораторной работе	12
11	Проработка и повторение лекционного материала. Подготовка к лабораторной работе	12
12	Проработка и повторение лекционного материала. Подготовка к лабораторной работе	12
13	Проработка и повторение лекционного материала. Подготовка к лабораторной работе	14
	Подготовка к аттестации	4

6. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

Информационные технологии, используемые при осуществлении образовательного процесса по дисциплине:

– сбор, хранение, систематизация, обработка и представление учебной и научной информации;

- обработка различного рода информации с применением современных информационных технологий;
- самостоятельный поиск дополнительного учебного и научного материала, с использованием поисковых систем и сайтов сети Интернет, электронных энциклопедий и баз данных;
- использование электронной почты для рассылки и асинхронного общения, чата преподавателей и обучающихся, переписки и обсуждения возникших учебных проблем для синхронного взаимодействия
- дистанционные образовательные технологии (при необходимости).

Интерактивные и активные образовательные технологии

№ раздела (темы)	Вид занятия (Л, ПЗ, С, ЛР)	Используемые интерактивные образовательные технологии	Количество часов	
1,4	Л	Лекция-визуализация	8	1
2,5	ПР	Мозговая атака	6	-
6,8,10	ЛР	Опережающая самостоятельная работа	8	5

Практическая подготовка обучающихся, не предусмотрена.

7. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ (ОЦЕНОЧНЫЕ МАТЕРИАЛЫ) ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ, ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

Фонд оценочных средств (оценочные материалы) для текущего контроля успеваемости, промежуточной аттестации по дисциплине приводятся в приложении.

8. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

8.1. Основная литература

1. Огнева, М. В. Программирование на языке C++: практический курс : учебное пособие для вузов / М. В. Огнева, Е. В. Кудрина. — Москва : Издательство Юрайт, 2020. — 335 с. — (Высшее образование). — ISBN 978-5-534-05123-0. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/454165>.
2. Зыков, С. В. Программирование. Объектно-ориентированный подход : учебник и практикум для вузов / С. В. Зыков. — Москва : Издательство Юрайт, 2020. — 155 с. — (Высшее образование). — ISBN 978-5-534-00850-0. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/451488>.
3. Подбельский, В. В. Программирование. Базовый курс C# : учебник для вузов / В. В. Подбельский. — Москва : Издательство Юрайт, 2020. — 369 с. — (Высшее образование). — ISBN 978-5-534-10616-9. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/450868>.

8.2. Дополнительная литература

1. Программирование: математическая логика : учебное пособие для вузов / М. В. Швецкий, М. В. Демидов, А. В. Голанова, И. А. Кудрявцева. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2020. — 675 с. — (Высшее образование). — ISBN 978-5-534-11009-8. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/439066>.
2. Зыков, С. В. Программирование : учебник и практикум для вузов / С. В. Зыков. — Москва : Издательство Юрайт, 2020. — 320 с. — (Высшее образование). — ISBN 978-5-

534-02444-9. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/450832>.

3. Зыков, С. В. Программирование. Функциональный подход : учебник и практикум для вузов / С. В. Зыков. — Москва : Издательство Юрайт, 2020. — 164 с. — (Высшее образование). — ISBN 978-5-534-00844-9. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/451972>.

8.3 Программное обеспечение

1. MSOffice,
2. AcademicEditionNetworked (Visual Studio Enterprise 2019).

8.4 Профессиональные базы данных

1. Электронная библиотечная система «СКСИ» [Электронный ресурс] – Режим доступа: <https://www.sksi.ru/environment/ebs/1363/>
2. Интернет университет информационных технологий [Электронный ресурс] – Режим доступа: <http://www.intuit.ru/>
3. Электронная библиотечная система «Urait» [Электронный ресурс] – Режим доступа: <https://urait.ru/>
4. Электронная библиотека информационных технологий CITForum.ru [Электронный ресурс] – Режим доступа: (<http://citforum.ru>).
5. Виртуальная академия Microsoft [Электронный ресурс] – Режим доступа (<http://aka.ms/studentcourse>).

8.5 Информационные справочные системы

1. Информационно-справочная система для программистов [Электронный ресурс] – Режим доступа: <http://life-prog.ru>

8.6 Интернет-ресурсы

1. Академия ORACLE [Электронный ресурс] – Режим доступа: <https://academy.oracle.com/ru/>
2. Научная сеть Scipeople [Электронный ресурс] – Режим доступа: <http://scipeople.ru/>
3. Портал открытых данных [Электронный ресурс] – Режим доступа <https://data.gov.ru/>.

8.7. Методические указания по освоению дисциплины.

Методические указания при работе над конспектом во время проведения лекции

В ходе лекционных занятий необходимо вести конспектирование учебного материала. Общие и утвердившиеся в практике правила и приемы конспектирования лекций:

Конспектирование лекций ведется в специально отведенной для этого тетради, каждый лист которой должен иметь поля, на которых делаются пометки из рекомендованной литературы, дополняющие материал прослушанной лекции, а также подчеркивающие особую важность тех или иных теоретических положений.

Необходимо записывать тему и план лекций, рекомендуемую литературу к теме. Записи разделов лекции должны иметь заголовки, подзаголовки, красные строки. Для выделения разделов, выводов, определений, основных идей можно использовать цветные карандаши и фломастеры.

Названные в лекции ссылки на первоисточники надо пометить на полях, чтобы при самостоятельной работе найти и вписать их.

В конспекте дословно записываются определения понятий, категорий и законов. Остальное должно быть записано своими словами.

Каждому обучающемуся необходимо выработать и использовать допустимые сокращения наиболее распространенных терминов и понятий.

В конспект следует заносить всё, что преподаватель пишет на доске, а также рекомендуемые схемы, таблицы, диаграммы и т.д.

Методические указания по подготовке к практическим работам

Целью практических и лабораторных работ является углубление и закрепление теоретических знаний, полученных обучающимися на лекциях и в процессе самостоятельного изучения учебного материала, а, следовательно, формирование у них определенных умений и навыков.

В ходе подготовки к практическим и лабораторным работам необходимо прочитать конспект лекции, изучить основную литературу, ознакомиться с дополнительной литературой, выполнить выданные преподавателем задания. При этом учесть рекомендации преподавателя и требования программы. Дорабатывать свой конспект лекции, делая в нем соответствующие записи из литературы. Желательно при подготовке к практическим и лабораторным работам по дисциплине одновременно использовать несколько источников, раскрывающих заданные вопросы.

Методические указания по организации самостоятельной работы

Самостоятельная работа приводит обучающегося к получению нового знания, упорядочению и углублению имеющихся знаний, формированию у него профессиональных навыков и умений.

Самостоятельная работа выполняет ряд функций:

- развивающую;
- информационно-обучающую;
- ориентирующую и стимулирующую;
- воспитывающую;
- исследовательскую.

Виды самостоятельной работы, выполняемые в рамках курса:

1. Проработка и повторение лекционного материала
2. Подготовка к практическим занятиям
3. Подготовка к лабораторным занятиям
4. Реферат
5. Подготовка к аттестации

Обучающимся рекомендуется с самого начала освоения курса работать с литературой и предлагаемыми заданиями в форме подготовки к очередному аудиторному занятию. При этом актуализируются имеющиеся знания, а также создается база для усвоения нового материала, возникают вопросы, ответы на которые обучающийся получает в аудитории.

Можно отметить, что некоторые задания для самостоятельной работы по курсу имеют определенную специфику. При освоении курса обучающийся может пользоваться библиотекой вуза, которая в полной мере обеспечена соответствующей литературой. Значительную помощь в подготовке к очередному занятию может оказать имеющийся в учебно-методическом комплексе краткий конспект лекций. Он же может использоваться и для закрепления полученного в аудитории материала.

Методические указания по работе с литературой

Всю литературу можно разделить на учебники и учебные пособия, оригинальные научные монографические источники, научные публикации в периодической печати. Из них можно выделить литературу основную (рекомендуемую), дополнительную и литературу для углубленного изучения дисциплины.

Изучение дисциплины следует начинать с учебника, поскольку учебник – это книга, в которой изложены основы научных знаний по определенному предмету в соответствии с целями и задачами обучения, установленными программой.

При работе с литературой следует учитывать, что имеются различные виды чтения, и каждый из них используется на определенных этапах освоения материала.

Предварительное чтение направлено на выявление в тексте незнакомых терминов и поиск их значения в справочной литературе. В частности, при чтении указанной литературы необходимо подробнейшим образом анализировать понятия.

Сквозное чтение предполагает прочтение материала от начала до конца. Сквозное чтение литературы из приведенного списка дает возможность обучающемуся сформировать свод основных понятий из изучаемой области и свободно владеть ими.

Выборочное – наоборот, имеет целью поиск и отбор материала. В рамках данного курса выборочное чтение, как способ освоения содержания курса, должно использоваться при подготовке к лабораторным практикумам по соответствующим разделам.

Аналитическое чтение – это критический разбор текста с последующим его конспектированием. Освоение указанных понятий будет наиболее эффективным в том случае, если при чтении текстов обучающийся будет задавать к этим текстам вопросы. Часть из этих вопросов сформулирована в приведенном в ФОС перечне вопросов для собеседования. Перечень этих вопросов ограничен, поэтому важно не только содержание вопросов, но сам принцип освоения литературы с помощью вопросов к текстам.

Целью изучающего чтения является глубокое и всестороннее понимание учебной информации.

Есть несколько приемов изучающего чтения:

1. Чтение по алгоритму предполагает разбиение информации на блоки: название; автор; источник; основная идея текста; фактический материал; анализ текста путем сопоставления имеющихся точек зрения по рассматриваемым вопросам; новизна.

2. Прием постановки вопросов к тексту имеет следующий алгоритм:

- медленно прочитать текст, стараясь понять смысл изложенного;
- выделить ключевые слова в тексте;
- постараться понять основные идеи, подтекст и общий замысел автора.

3. Прием тезирования заключается в формулировании тезисов в виде положений, утверждений, выводов.

К этому можно добавить и иные приемы: прием реферирования, прием комментирования.

Важной составляющей любого солидного научного издания является список литературы, на которую ссылается автор. При возникновении интереса к какой-то обсуждаемой в тексте проблеме всегда есть возможность обратиться к списку относящейся к ней литературы. В этом случае вся проблема как бы разбивается на составляющие части, каждая из которых может изучаться отдельно от других. При этом важно не терять из вида общий контекст и не погружаться чрезмерно в детали, потому что таким образом можно не увидеть главного.

9. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Для реализации дисциплины требуется следующее материально-техническое обеспечение (специальные помещения):

- для проведения занятий лекционного типа
учебная аудитория, оснащенная учебной мебелью, оборудованная проектором, ПК, экраном, доской.
- для проведения занятий семинарского типа, практических занятий
учебная аудитория, оснащенная учебной мебелью, оборудованная проектором, ПК, экраном, доской.
- для проведения, текущего контроля и промежуточной аттестации
учебная аудитория, оснащенная учебной мебелью, оборудованная проектором, ПК, экраном, доской.
- для групповых и индивидуальных консультаций

учебная аудитория, оснащенная учебной мебелью, оборудованная проектором, ПК, экраном, доской.

- для самостоятельной работы:

помещение, оснащенное компьютерной техникой с возможностью подключения к сети "Интернет" и обеспечением доступа в электронную информационно-образовательную среду Института

10. ОСОБЕННОСТИ ОСВОЕНИЯ ДИСЦИПЛИНЫ ЛИЦАМИ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ

Обучающимся с ограниченными возможностями здоровья предоставляются специальные учебники, учебные пособия и дидактические материалы, специальные технические средства обучения коллективного и индивидуального пользования, услуги ассистента (тьютора), оказывающего обучающимся необходимую техническую помощь, а также услуги сурдопереводчиков и тифлосурдопереводчиков.

Освоение дисциплины обучающимися с ограниченными возможностями здоровья может быть организовано совместно с другими обучающимися, а также в отдельных группах.

Освоение дисциплины обучающимися с ограниченными возможностями здоровья осуществляется с учетом особенностей психофизического развития, индивидуальных возможностей и состояния здоровья.

В целях доступности получения высшего образования по образовательной программе лицами с ограниченными возможностями здоровья при освоении дисциплины обеспечивается:

1) для лиц с ограниченными возможностями здоровья по зрению:

– присутствие тьютора, оказывающий студенту необходимую техническую помощь с учетом индивидуальных особенностей (помогает занять рабочее место, передвигаться, прочесть и оформить задание, в том числе, записывая под диктовку),

– письменные задания, а также инструкции о порядке их выполнения оформляются увеличенным шрифтом,

– специальные учебники, учебные пособия и дидактические материалы (имеющие крупный шрифт или аудиофайлы),

– индивидуальное равномерное освещение не менее 300 люкс,

– при необходимости студенту для выполнения задания предоставляется увеличивающее устройство;

2) для лиц с ограниченными возможностями здоровья по слуху:

– присутствие ассистента, оказывающий студенту необходимую техническую помощь с учетом индивидуальных особенностей (помогает занять рабочее место, передвигаться, прочесть и оформить задание, в том числе, записывая под диктовку),

– обеспечивается наличие звукоусиливающей аппаратуры коллективного пользования, при необходимости обучающемуся предоставляется звукоусиливающая аппаратура индивидуального пользования;

– обеспечивается надлежащими звуковыми средствами воспроизведения информации;

3) для лиц с ограниченными возможностями здоровья, имеющих нарушения опорно-двигательного аппарата:

– письменные задания выполняются на компьютере со специализированным программным обеспечением или надиктовываются тьютору;

– по желанию студента задания могут выполняться в устной форме.

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ (ОЦЕНОЧНЫЕ МАТЕРИАЛЫ) ДЛЯ
ПРОВЕДЕНИЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ И
ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ**

по дисциплине «Технологии программирования»

1. Показатели, критерии оценки освоения дисциплины

Результаты обучения	Показатели оценивания	Критерии оценивания	Процедуры оценивания
ОПК-6 Способен разрабатывать алгоритмы и программы, пригодные для практического применения в области информационных систем и технологий			
<p>Знает содержание, объекты и субъекты информационного общества, критерии эффективности его функционирования; структуру интеллектуального капитала, проблемы инвестиций в экономику информатизации и методы оценки эффективности; правовые, экономические, социальные и психологические аспекты информатизации; теоретические проблемы прикладной информатики, в том числе семантической обработки информации, развитие представлений об оценке качества информации в информационных системах; современные методы,</p>	<p>Демонстрация знаний в области знаний по способам разработки алгоритмов и программ</p>	<p>Полнота и качество знаний в области знаний по способам разработки алгоритмов и программ</p>	<p>устный опрос</p>

средства, стандарты информатики для решения прикладных задач различных классов; правовые, экономические, социальные и психологические аспекты информатизации деятельности организационно-экономических систем.			
Умеет проводить анализ современных методов и средств информатики для решения прикладных задач различных классов.	Демонстрация действий в области знаний по способам разработки алгоритмов и программ	Полнота и правильность действий в области знаний по способам разработки алгоритмов и программ	Практические задания
Владеет навыками применения современных программно-технических средств для решения прикладных задач различных классов.	Демонстрация навыков в области знаний по способам разработки алгоритмов и программ	Полнота и правильность действий, наличие навыков в области знаний по способам разработки алгоритмов и программ	Практические задания
ОПК-6			Промежуточная аттестация: дифференцированный зачет

2. Методические материалы, определяющие процедуры оценивания

2.1. Методические материалы, определяющие процедуры оценивания в рамках текущего контроля успеваемости

Устные опросы проводятся во время лекций, практических занятий и возможны при проведении промежуточной аттестации в качестве дополнительного испытания при недостаточности результатов тестирования. Основные вопросы для устного опроса доводятся до сведения студентов на предыдущем занятии.

Количество вопросов определяется преподавателем.

Время проведения опроса от 10 минут до 1 академического часа.

Устные опросы строятся так, чтобы вовлечь в тему обсуждения максимальное количество обучающихся в группе, проводить параллели с уже пройденным учебным материалом данной дисциплины и смежными курсами, находить удачные примеры из современной действительности, что увеличивает эффективность усвоения материала на ассоциациях.

Критерии и шкала оценки устного опроса

Развернутый ответ студента должен представлять собой связное, логически последовательное сообщение на заданную тему, показывать его умение применять определения, правила в конкретных случаях.

«отлично» ставится, если:

- 1) студент полно излагает материал, дает правильное определение основных понятий;
- 2) обнаруживает понимание материала, может обосновать свои суждения, применить знания на практике, привести необходимые примеры не только из учебника, но и самостоятельно составленные;
- 3) излагает материал последовательно и правильно с точки зрения норм литературного языка.

«хорошо» - студент дает ответ, удовлетворяющий тем же требованиям, что и для «отлично», но допускает 1–2 ошибки, которые сам же исправляет, и 1–2 недочета в последовательности и языковом оформлении излагаемого.

«удовлетворительно» – студент обнаруживает знание и понимание основных положений данной темы, но:

- 1) излагает материал неполно и допускает неточности в определении понятий или формулировке правил;
- 2) не умеет достаточно глубоко и доказательно обосновать свои суждения и привести свои примеры;
- 3) излагает материал непоследовательно и допускает ошибки в языковом оформлении излагаемого.

«неудовлетворительно» ставится, если студент обнаруживает незнание большей части соответствующего вопроса, допускает ошибки в формулировке определений и правил, искажающие их смысл, беспорядочно и неуверенно излагает материал. Оценка «2» отмечает такие недостатки в подготовке, которые являются серьезным препятствием к успешному овладению последующим материалом.

Практические задания выполняются студентами на практических занятиях. Студентам необходимо выполнить практические задания, указанные преподавателем. Результаты работы сохранить в файлах. После выполнения заданий необходимо преподавателю продемонстрировать результаты работы и быть готовым ответить на вопросы и продемонстрировать выполнение отдельных пунктов заданий. Защита выполненных практических заданий осуществляется на практическом занятии.

Критерии и шкала оценки практических заданий

«отлично» ставится, если: студент самостоятельно и правильно решил учебно-профессиональную задачу, уверенно, логично, последовательно и аргументировано излагал свое решение, используя изученные понятия.

«хорошо» ставится, если: студент самостоятельно и в основном правильно решил учебно-профессиональную задачу, уверенно, логично, последовательно и аргументировано излагал свое решение, используя изученные понятия.

«удовлетворительно» ставится, если: студент в основном решил учебно-профессиональную задачу, допустил несущественные ошибки, слабо аргументировал свое решение, используя в основном изученные понятия.

«неудовлетворительно» ставится, если: студент не решил учебно-профессиональную задачу.

2.2. Методические материалы, определяющие процедуры оценивания в рамках промежуточной аттестации

Промежуточная аттестация по дисциплине проводится в форме **дифференцированного зачета**.

Дифференцированный зачет служит формой проверки успешного выполнения студентами практических заданий и усвоения знаний на занятиях лекционного и семинарского типа.

Результаты дифференцированного зачета определяются на основании результатов текущего контроля успеваемости студента в течение периода обучения.

Критерии и шкала оценки дифференцированного зачета

Дифференцированный зачет выставляется по совокупному результату текущего контроля успеваемости студента по дисциплине:

«**отлично**» - выставляется при условии, если студент выполнил все практические занятия с оценкой не ниже «хорошо и отлично», результаты тестирования и устного опроса в совокупности также оценены с оценкой не ниже «хорошо и отлично» при этом совокупный средний балл оценок не ниже 4,6.

«**хорошо**» - выставляется при условии, если студент выполнил все практические занятия с оценкой не ниже «хорошо, отлично, удовлетворительно», результаты тестирования и устного опроса в совокупности также оценены с оценкой не ниже «хорошо, отлично, удовлетворительно», при этом совокупный средний балл оценок не ниже 3,8.

«**удовлетворительно**» - выставляется при условии, если студент выполнил все практические занятия с оценкой не ниже «удовлетворительно», результаты тестирования и устного опроса в совокупности также оценены с оценкой не ниже «удовлетворительно», при этом совокупный средний балл оценок ниже 3,8.

«**неудовлетворительно**» - не выполнены условия для получения оценки «удовлетворительно».

3. Типовые контрольные задания

Типовые задания для текущего контроля успеваемости

3.1 Перечень типовых вопросов для устного опроса

Тема 1. Надежное программное средство как продукт технологии программирования.

- Исторический и социальный контекст программирования
- Программа как формализованное описание процесса обработки данных.
- Программное средство.
- Неконструктивность понятия правильной программы.
- Надежность программного средства.
- Технология программирования как технология разработки надежных программных средств.
- Технология программирования и информатизация общества.

Тема 2. Источники ошибок в программных средствах

- Интеллектуальные возможности человека.
- Неправильный перевод как причина ошибок в программных средствах.
- Модель перевода.
- Основные пути борьбы с ошибками.

Тема 3. Общие принципы разработки программных средств

- Специфика разработки программных средств.
- Жизненный цикл программного средства.
- Понятие качества программного средства.
- Обеспечение надежности - основной мотив разработки программных средств.
- Методы борьбы со сложностью.
- Обеспечение точности перевода.
- Преодоление барьера между пользователем и разработчиком.
- Контроль принимаемых решений.

Тема 4. Внешнее описание программного средства

- Назначение внешнего описания программного средства и его роль в обеспечении качества программного средства.
- Определение требований к программному средству.
- Спецификация качества программного средства.
- Функциональная спецификация программного средства.
- Методы контроля внешнего описания программного средства.

Тема 5. Методы спецификации семантики функций

- Основные подходы к спецификации семантики функций.
- Метод таблиц решений.
- Операционная семантика.
- Денотационная семантика.
- Аксиоматическая семантика.
- Языки спецификаций.

Тема 6. Архитектура программного средства

- Понятие архитектуры программного средства.
- Основные классы архитектур программных средств.
- Архитектурные функции.
- Контроль архитектуры программных средств.

Тема 7. Разработка структуры программы и модульное программирование

- Цель модульного программирования.
- Основные характеристики программного модуля.
- Методы разработки структуры программы.
- Контроль структуры программы.

Тема 8. Разработка программного модуля

- Порядок разработки программного модуля.
- Структурное программирование.
- Пошаговая детализация и понятие о псевдокоде.
- Контроль программного модуля.

Тема 9. Доказательство свойств программ

- Обоснования программ.
- Формализация свойств программ.
- Свойства простых операторов.
- Завершимость выполнения программы.
- Пример доказательства свойства программы.

Тема 10. Тестирование и отладка программного средства

- Основные понятия.
- Принципы и виды отладки программного средства.
- Заповеди отладки программного средства.
- Автономная отладка программного средства.

- Комплексная отладка программного средства.
- Тема 11. Обеспечение функциональности и надежности программного средства
- Функциональность и надежность как обязательные критерии качества программного средства.
 - Обеспечение завершенности программного средства.
 - Обеспечение точности программного средства.
 - Обеспечение автономности программного средства.
 - Обеспечение устойчивости программного средства.
 - Обеспечение защищенности программных средств.
- Тема 12. Обеспечение качества программного средства
- Общая характеристика процесса обеспечения качества программного средства.
 - Обеспечение легкости применения программного средства.
 - Обеспечение эффективности программного средства.
 - Обеспечение сопровождаемости программного средства.
 - Обеспечение мобильности.
- Тема 13. Документирование программных средств
- Документация, создаваемая и используемая в процессе разработки программных средств.
 - Пользовательская документация программных средств.
 - Документация по сопровождению программных средств.

3.2. Примеры типовых практических заданий

Пример практического задания:

Практическое занятие № 1.

Динамическая память в языке C++

1. Цель работы: Закрепление знаний об управлении оперативной памятью в C/C++ : динамическая память в C (функции malloc и free) , динамическая память в C++ (функции new и delete), многофайловые проекты.

2. Основные сведения

Для выделения и освобождения динамической памяти в C используются функции malloc() и free(), которые содержатся в заголовочном файле stdlib.h. Прототип функции для выделения памяти:

```
void *malloc(int size);
```

Пример использования этих функций:

```
//dynam.cpp динамическая память
```

```
#include<stdio.h>
```

```
#include<stdlib.h>
```

```
#include<conio.h>
```

```
main() {
```

```
int i,j,k;
```

```
void *p;
```

```
struct sotrud {
```

```
char *fio[20];
```

```
float oklad;
```

```
struct sotrud *p ; } a,b,c;
```

```
struct sotrud *p1,*p2;
```

```
p1=(sotrud*)malloc(sizeof(sotrud));
```

```

if(!p1) {printf("Недост. памяти\n");exit(1);}
p1->oklad=5.0;*p1->fio="Иванов";
p2=p1;
printf("Оклад %4.1f ФИО %s\n",p2->oklad,*p2->fio);
free(p1);
getch(); return 0; }

```

Указателю *p в структуре sotrud можно присвоить значение указателя на следующую (или предыдущую) структуру, тогда получится односвязный список (или стек). Можно в структуру ввести 2 указателя на предыдущую и последующую структуры, получится двухсвязный список.

В C++ для динамического выделения и освобождения памяти применяются операции new и delete . Синтаксис применения операции new:

```
<указатель> = new <имя типа> <инициализатор>
```

Необязательный инициализатор дает возможность присвоить начальное значение.

Синтаксис применения операции delete: delete <указатель>

При этом освобождается память, выделенная для данного указателя. Если указателю присваивалось несколько значений, берется последнее.

Выделять динамически память под простые переменные невыгодно, т.к. на указатель расходуется память, сравнимая по величине с самой переменной. Более выгодно выделять динамически память под массивы и типы, определенные пользователем, например, структуры.

Замечательным признаком динамического выделения памяти под массивы является то, что мы можем заранее не знать размера массива, под который следует выделить память, размер массива можно задать переменной целого типа.

Приведем пример программы выделения память под массив, ввода элементов массива, определения суммы элементов массива.

```

//dyn_mass динамический массив
#include<iostream.h>
#include<conio.h>
main(){int i,s,n,*p;
cout<<"Enter the size of the array\n";
cin>>n;
p=new int[n];
if(!p) {cout<<"Недостаточно памяти\n";exit(1); }
s=0;
for(i=0;i<n;i++){
cout<<"Enter the "<<i<<" element of the array\n";
cin>>p[i];//Ввод элементов массива
s+=p[i];// накопление суммы
}
cout<<"The sum of the elements "<<s;
getch();
return 0;
}

```

3. Выполнение работы

Набрать и откомпилировать приведенный пример программы. Объяснить работу программы.

Согласно вариантам заданий составить программу на C++: задать размер массива (от 5 до 7), создать динамический массив, ввести его элементы, провести расчет согласно варианту и вывести результаты расчета.

4. Варианты заданий

1. Найти произведение положительных элементов массива.
2. Найти сумму отрицательных элементов массива.
3. Найти число положительных элементов.
4. Найти число элементов, которые больше 2 и меньше 10
5. Найти сумму положительных элементов.
6. Найти сумму квадратов элементов.
7. Найти сумму элементов, больших 3.
8. Найти сумму элементов, меньших по модулю 5.
9. Найти сумму кубов элементов.
10. Найти число элементов, меньших 10.

5. Контрольные вопросы

1. Опишите функции `malloc()` и `free()`. Какие заголовочные файлы следует включить в программу для использования этих функций?
2. Опишите операции `new` и `delete`. Как следует использовать в программе эти операции? Нужны ли для этого заголовочные файлы?
3. Как выделить память под массив переменных определенного типа?
4. Можно ли не знать заранее размер динамического массива?
6. Литература: [1], глава 3; [2], семинар 9.

Типовые задания для промежуточной аттестации

3.3. Типовые контрольные вопросы для устного опроса на дифференцированном зачете

1. Обеспечение защищенности программных средств.
2. Технология программирования и информатизация общества.
3. Принципы и виды отладки программного средства.
4. Заповеди отладки программного средства.
5. Общие принципы разработки программных средств
6. Специфика разработки программных средств.
7. Внешнее описание программного средства
8. Назначение внешнего описания программного средства и его роль в обеспечении качества программного средства.
9. Функциональная спецификация программного средства.
10. Методы спецификации семантики функций
11. Общая характеристика процесса обеспечения качества программного средства.
12. Понятие архитектуры программного средства.
13. Основные классы архитектур программных средств.
14. Архитектурные функции.
15. Документация, создаваемая и используемая в процессе разработки программных средств.
16. Пользовательская документация программных средств.
17. Цель модульного программирования.
18. Основные характеристики программного модуля.
19. Методы разработки структуры программы.
20. Автономная отладка программного средства.

21. Источники ошибок в программных средствах. Основные пути борьбы с ошибками.
22. Тестирование и отладка программного средства. Основные понятия
23. Комплексная отладка программного средства.
24. Функциональность и надежность как обязательные критерии качества программного средства.
25. Надежное программное средство как продукт технологии программирования. Программное средство.
26. Технология программирования как технология разработки надежных программных средств.
27. Жизненный цикл программного средства.
28. Понятие качества программного средства.
29. Обеспечение точности программного средства.
30. Обеспечение эффективности программного средства.
31. Документация по сопровождению программных средств.

**Дополнения и изменения к рабочей программе дисциплины
«Технологии программирования»
направление подготовки 09.03.02 Информационные системы и технологии
направленность (профиль) программы: Проектирование информационных систем и их компонентов
год начала подготовки 2018, 2019, 2020 г.**

Внесенные изменения на 2022/2023 учебный
год



Ж.В. Игнатенко
«20» _____ 2022 г.

В рабочую программу вносятся следующие изменения:

1) 8.1 Основная литература

1. Кулямин, В. В. Технологии программирования. Компонентный подход : учебное пособие / В. В. Кулямин. — 3-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. — 590 с. — ISBN 978-5-4497-0884-7. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/102071.html>

2. Гниденко, И. Г. Технологии и методы программирования : учебное пособие для вузов / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — Москва : Издательство Юрайт, 2022. — 235 с. — (Высшее образование). — ISBN 978-5-534-02816-4. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/489920>

3. Зыков, С. В. Программирование. Объектно-ориентированный подход : учебник и практикум для вузов / С. В. Зыков. — Москва : Издательство Юрайт, 2022. — 155 с. — (Высшее образование). — ISBN 978-5-534-00850-0. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/490423>

2) 8.2 Дополнительная литература

1. Минакова, О. В. Технологии программирования: паттерны проектирования в реализации JavaFX приложений : практикум / О. В. Минакова. — Воронеж : Воронежский государственный технический университет, ЭБС АСВ, 2020. — 156 с. — ISBN 978-5-7731-0911-2. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/111505.html>

2. Горелов, С. В. Современные технологии программирования: разработка Windows-приложений на языке C#. В 2 томах. Т.1 : учебник / С. В. Горелов ; под редакцией П. Б. Лукьянова. — Москва : Прометей, 2019. — 362 с. — ISBN 978-5-907100-09-1. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/94532.html>

3. Зыков, С. В. Программирование : учебник и практикум для вузов / С. В. Зыков. — Москва : Издательство Юрайт, 2022. — 320 с. — (Высшее образование). — ISBN 978-5-534-02444-9. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/489754>

3) 8.3. Программное обеспечение

Microsoft Windows, Яндекс 360, Microsoft Office Professional Plus 2019, Google Chrome, Яндекс.Браузер.

4) 8.5. Информационные справочные системы

1С: Библиотека - <https://www.sksi.ru/environment/eor/library/>

Справочно-правовая система «КонсультантПлюс» - <http://www.consultant.ru/>

Поисковые системы

Поисковая система Яндекс- <https://www.yandex.ru/>

Поисковая система Rambler – <https://www.rambler.ru/>

Рабочая программа пересмотрена и рекомендована на заседании кафедры прикладной информатики и математики от «19» мая 2022 г. протокол №9
зав. кафедрой _____ Ж.В. Игнатенко

Рабочая программа пересмотрена и одобрена на заседании учебно-методической комиссии ФИСТ от «20» мая 2022 г. протокол №9

Председатель УМК _____ Ж.В. Игнатенко

СОГЛАСОВАНО:

Зав. выпускающей кафедрой прикладной информатики и математики
_____ Ж.В. Игнатенко «20» мая 2022 г.