МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

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

У Т В Е Р Ж Д А Ю
Проректор по учебной работе
_____ П.Е. Троян
«14» 10 2016 г.

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

Программирование

Уровень основной образовательной программы бакалавриат

Направление подготовки 09.03.01 – Информатика и вычислительная техника.

Профиль(и) Программное обеспечение средств вычислительной техники и автоматизированных систем.

Форма обучения очная

Факультет систем управления

Кафедра автоматизированных систем управления

Курс 1

Семестр 1, 2

Учебные планы набора 2016 года и последующих лет.

Распределение рабочего времени:

Виды учебной работы	Семестр 1	Семестр 2	Всего	Единицы
Лекции	18	16	34	часов
Лабораторные работы	36	34	70	часов
Практические занятия	36	34	70	часов
Курсовой проект/работа (КРС) (аудиторная)	Не предусмотрено			часов
Всего аудиторных занятий	90	84	174	часов
из них в интерактивной форме	10	10	20	часов
Самостоятельная работа студентов (СРС)	90	96	186	часов
Всего (без экзамена)	180	180	360	часов
Самост. работа на подготовку и сдачу экзамена	–	36	36	часов
Общая трудоемкость	180	216	396	часов
(в зачетных единицах)	5	6	11	ЗЕТ

Зачёт 1 семестр

Экзамен 2 семестр

Томск 2016

Согласована на портале № 5770

Рассмотрена и одобрена на заседании кафедры
протокол № 2 от « 26 » 11 2016 г.

Рабочая программа по дисциплине составлена с учетом требований Федерального Государственного образовательного стандарта высшего профессионального образования (ФГОС ВО) по направлению подготовки 09.03.01 Информатика и вычислительная техника (квалификация (степень) "бакалавр"), утвержденного Приказом Министерства образования и науки Российской Федерации от 12 января 2016 г. № 5, рассмотрена и утверждена на заседании кафедры 12 февраля 2016 г., протокол № 5.

Разработчик к.т.н., доцент каф. АСУ _____ В.Д. Сибилёв

Зав. обеспечивающей кафедрой АСУ
д.т.н., профессор _____ А.М. Кориков

Рабочая программа согласована с факультетом, профилирующей и выпускающей кафедрами специальности.

Декан_ к.т.н., доцент _____ П.В. Сенченко

Заведующий профилирующей и выпускающей
кафедрой АСУ, д.т.н., профессор _____ А.М. Кориков

Эксперты:
Кафедра АСУ, доцент _____ А.И. Исакова

Согласована на портале № 5770

1. Цели и задачи дисциплины: приобретение теоретических знаний и практических навыков в области алгоритмизации и программирования на алгоритмических языках.

2. Место дисциплины в структуре ООП

Дисциплина Б1.Б.15 – Программирование входит в базовую часть профессионального цикла. Для успешного овладения дисциплиной желательно знакомство с основами информатики на уровне средней школы. Знания и навыки, приобретённые в результате изучения дисциплины, используются практически во всех дисциплинах профессионального цикла, определяемых учебным планом, а также при выполнении учебно-исследовательских работ и заданий по курсовому и дипломному проектированию.

3. Требования к результатам освоения дисциплины

Процесс изучения дисциплины «Программирование» направлен на формирование следующих компетенций:

ОПК-2 Способность осваивать методики использования программных средств для решения практических задач.

ПК-1 Способность разрабатывать модели компонентов информационных систем, включая модели баз данных и модели интерфейсов «человек – электронно-вычислительная машина».

В результате освоения содержания дисциплины «Программирование» студент должен:

Знать:

- современные методы и средства разработки алгоритмов и программ,
- способы записи алгоритмов на языке высокого уровня,
- способы отладки, тестирования и документирования программ.

Уметь:

- разрабатывать алгоритмы решения задач,
- писать программы на языке С,
- выполнять отладку и тестирование программ,

Владеть:

- приёмами проектирования программ,
- приёмами структурного программирования,
- методами верификации программ,
- приёмами отладки и тестирования программ.

4. Объем дисциплины и виды учебной работы

Общая трудоемкость дисциплины составляет 11 зачетных единиц.

Вид учебной работы	Всего часов	Семестры	
		1	2
Аудиторные занятия (всего)	**174**	**90**	**84**
В том числе:			
Лекции	34	18	16
Лабораторные работы (ЛР)	70	36	34
Практические занятия (ПЗ)	70	36	34
Семинары (С)	–	–	–
Курсовой проект/(работа) (аудиторная нагрузка)	–	–	–
Самостоятельная работа (всего)	**186**	**90**	**96**
В том числе:			
Курсовой проект (работа) (самостоятельно)	–	–	–
Расчетно-графические работы	–	–	–
Проработка лекционного материала	12	5	7
Подготовка к лабораторным работам	72	36	36
Подготовка к практическим занятиям	72	36	36
Самостоятельное изучение тем теоретической части	30	13	17
Подготовка к экзамену	36	–	36
Вид промежуточной аттестации (зачет, экзамен)		Зачёт	Экзамен
Общая трудоемкость час	**396**	**180**	**216**
зач. ед. (до сотых долей)	**11**	**5**	**6**

Согласована на портале № 5770

5. Содержание дисциплины

5.1. Разделы дисциплины и виды занятий

Таблица 5.1

№ п/п	Наименование раздела дисциплины	Лекц.	Практ.	Лаб. зан.	Семин.	СРС	Всего час.	Формируемые компетенции (ОК, ПК, ПСК)
1	2	3	4	5	6	7	8	9
1.	Основы алгоритмизации и программирования. (Семестр 1)	18	36	36	–	90	180	ОПК-2, ПК-1
2.	Язык программирования Си. (Семестр 2)	16	34	34		96	180	ОПК-2, ПК-1
	Всего	**34**	**70**	**70**	**–**	**186**	**360**	

5.2. Содержание разделов дисциплины (по лекциям)

Таблица 5.2

№п/п	Наименование раздела	Содержание раздела	Трудоемкость (час.)	Формируемые компетенции (ОК, ПК, ПСК)
1	2	3	4	5
1.	**Основы алгоритмизации и программирования.**	**Тема 1. Введение.** История развития ЭВМ и программирования. Понятие алгоритма. Понятие программы. Организация обработки и исполнения программы.	2	ОПК-2, ПК-1
		Тема 2. Структурное программирование Принципы и конструкции структурного программирования. Пошаговая детализация. Способы записи алгоритмов.	2	
		Тема 3. Основные понятия алгоритмических языков программирования. Алфавит. Лексема. Тип данных. Константа. Переменная. Операция. Разделитель. Выражение. Оператор. Типы операторов. Блок операторов. Подпрограмма.	2	
		Тема 4. Рекуррентные алгоритмы и рекурсия. Понятие рекуррентного алгоритма и рекуррентной последовательности. Задача вычисления элемента последовательности с заданным номером. Вычисление суммы конечного числа элементов. Вычисление бесконечных сумм. Понятие рекурсии. Примеры рекурсивных алгоритмов. Прямая и косвенная рекурсии. Преимущества и недостатки рекурсивного описания алгоритмов.	2	
		Тема 5. Алгоритмы нахождения корней функции. Принципы численного нахождения корня функции. Метод дихотомии. Метод Ньютона. Метод хорд. Метод итераций. Оценка трудоёмкости алгоритмов.	2	
		Тема 6. Проверка правильности алгоритмов. Основные источники ошибок и методы борьбы с ними. Тестирование. Методы верификации алгоритмов. Защита от неправильных данных.	2	
		Тема 7. Подпрограммы в языках программирования. Понятие подпрограммы. Процедуры и функции. Формальные и фактические параметры подпрограмм. Описание и вызов процедур и функций. Роль подпрограмм в процессе написания и отладки программы.	2	
		Тема 8. Алгоритмы поиска и сортировки. Общая постановка задачи поиска и её разновидности. Поиск в неупорядоченном и упорядоченном массивах. Общая постановка задачи сортировки. Простые методы сортировки массивов: сортировка включением, сортировка выбором, сортировка обменом. Алгоритм сортировки Шелла. Алгоритм сортировки Хоара.	4	
2.	**Язык программирования Си.**	**Тема 1. Препроцессорные средства.** Назначение препроцессора. Стадии препроцессорной обработки. Директивы препроцессора. Замены в тексте. Включение текстов из файлов. Условная компиляция. Макроподстановки. Встроенные макроимена.	2	ОПК-2, ПК-1
		Тема 2. Указатели.	2	

Согласована на портале № 5770

		Понятия объекта, указателя на объект. Операции над указателями. Указатели и массивы. Моделирование динамических массивов. Моделирование многомерных массивов.		
		Тема 3. Обработка символьных данных. Понятие символьной строки. Принцип лексикографического порядка. Сравнение строк. Поиск в упорядоченном массиве строк. Поиск подстроки.	2	
		Тема 4. Функции. Определение, прототип и вызов функции. Параметры функции. Указатели на функции. Функции с переменным количеством параметров.	2	
		Тема 5. Структуры и объединения. Понятие структуры. Определение структуры и выделение памяти. Инициализация и присваивание структур. Доступ к полям структуры. Структуры, массивы и указатели. Динамические информационные структуры.	2	
		Тема 6. Ввод и вывод. Понятия потока и файла. Потоковый ввод/вывод.	2	
		Тема 7. Обработка графических объектов. Графическая библиотека Open GL. Установка и настройка Open GL. Инициализация рабочей области. Типы данных и формат команд. Графические примитивы. Модельные преобразования.	4	

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

Обеспечивающие (предыдущие) дисциплины не предусмотрены.

№ п/п	Наименование обеспечиваемых (последующих) дисциплин	№ № разделов данной дисциплины, которые обеспечивают изучение последующих дисциплин	
		1	**2**
1.	Б1.Б.18 Математическая логика и теория алгоритмов	+	+
2.	Б1.В.ОД.7 Объектно-ориентированное программирование		+
3.	Б1.В.ОД.8 Структуры и алгоритмы обработки данных в ЭВМ	+	+
4.	Б1.В.ОД.9 Основы разработки программного обеспечения	+	
5.	Б1.В.ОД.12 Системы цифровой обработки сигналов	+	+
6.	Б3.В.ДВ.1 GRID-технологии / Параллельное программирование		+

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

Перечень компетенций	Виды занятий				Формы контроля *(примеры)*
	Л	Пр	Лаб	СРС	
ОПК-2	+	+	+	+	Тест, конспект, устный ответ на занятии, контрольная работа, индивидуальное задание.
ПК-1		+	+	+	Отчёт о лабораторной работе, контрольная работа, индивидуальное задание.

Л – лекция, Пр – практические и семинарские занятия, Лаб – лабораторные работы, СРС – самостоятельная работа студента

6. Методы и формы организации обучения

Для успешного освоения дисциплины применяются различные образовательные технологии, которые обеспечивают достижение планируемых результатов обучения согласно основной образовательной программе, с учётом требований к объёму занятий в интерактивной форме.

Технологии интерактивного обучения при разных формах занятий

Формы / Методы	Лекции	Лабораторные работы	Практические /семинарские занятия (час)	Всего (час)
«Мозговой штурм»			**4**	**4**
Игра	2			**2**
Работа в команде		4	**2**	**6**
Разработка проекта		4	**4**	**10**
Итого интерактивных занятий	**2**	**8**	**10**	**20**

Интерактивное обучение реализуется на практических занятиях.

Согласована на портале № 5770

«Мозговой штурм» используется на начальных этапах разработки алгоритмов для уяснения задачи и генерации идеи алгоритма.

Работа в команде применяется при решении усложнённых задач. Группа студентов формируется в команду по добровольному принципу. Команда должна выбрать лидера, уяснить задачу, выделить подзадачи и распределить их между членами команды. Результатом работы команды должна быть программа, решающая задачу. Методика направлена на развитие чувства ответственности и умения согласовывать свои решения и действия с действиями других членов команды.

Разработка проекта используется во втором семестре обучения. Методика направлена на выработку навыков группового проектирования программных средств.

7. Лабораторный практикум

№ п/п	№ раздела дисциплины из табл. 5.1	Наименование лабораторных работ	Трудоёмкость (час.)	Компетенции ОК, ПК, ПСК
1.	**1** (Семестр 1)	Знакомство с интегрированной средой программирования Code::Block	4	ОПК-2, ПК-1
2.		Изучение управляющих конструкций структурного программирования.	8	ОПК-2, ПК-1
3.		Реализация рекуррентных алгоритмов. Часть I. Табулирование функций, заданных рядом Тейлора.	8	ОПК-2, ПК-1
4.		Реализация рекуррентных алгоритмов. Часть II. Приближённое решение уравнений.	8	
5.		Реализация алгоритмов сортировки и поиска.	8	ОПК-2, ПК-1
6.	**2** (Семестр 2)	Обработка динамических массивов.	6	
7.		Обработка символьных данных.	4	
8.		Обработка структур.	8	
9.		Обработка файлов.	8	
10.		Обработка графических объектов.	8	

8. Практические занятия

№ п/п	№ раздела дисциплины из табл. 5.1	Наименование практических занятий	Трудоёмкость (час.)	Компетенции ОК, ПК, ПСК
1.	**1** (Семестр 1)	Алгоритмизация.	8	ОПК-2, ПК-1
2.		Форматный ввод/вывод данных.	2	ОПК-2, ПК-1
3.		Простейшие программные конструкции.	8	
4.		Решение типовых задач.	12	
5.		Обработка массивов.	6	
6.	**2** (Семестр 2)	Обработка символьной информации.	4	
7.		Обработка структур.	8	ОПК-2, ПК-1
8.		Обработка динамических массивов.	8	
9.		Потоковый ввод/вывод.	6	
10.		Обработка графических объектов.	8	

9. Самостоятельная работа

№ п/п	№ раздела дисциплины из табл. 5.1	Тематика самостоятельной работы	Трудоемкость (час.)	Компетенции ОК, ПК, ПСК	Контроль выполнения работы
1.	1, 2	Проработка лекционного материала	10	ОПК-2, ПК-1	Опрос на занятиях, тест, контрольная работа.
2.	1, 2	Подготовка к лабораторным работам.	72	ОПК-2, ПК-1	Отчёт, защита лаб. работы.
3.	1, 2	Подготовка к практическим занятиям.	72	ОПК-2, ПК-1	Защита программ, контрольная работа.
4.	1, 2	Самостоятельное изучение тем теоретической части	32	ПСК-10, 11	Дом. задание, тест.
5.	1, 2	Подготовка и сдача экзамена	36	ОПК-2, ПК-1	Оценка за экзамен.
			222		

Согласована на портале № 5770

Темы для самостоятельного изучения

1. Базовые понятия и синтаксис языка Си.
2. Классы памяти и организация Си-программ.
3. Ввод/вывод нижнего уровня.

10. Примерная тематика курсовых проектов (работ) – не предусмотрены.

11. Рейтинговая система для оценки успеваемости студентов

Для стимулирования планомерности работы студента в семестре в раскладку баллов по элементам контроля введён компонент своевременности представления и защиты отчётов о лабораторных работах.

Текущая успеваемость оценивается в баллах нарастающим итогом. Выполняется контроль

— усвоения теоретического материала – проведение трёх тестов;

— выполнения лабораторных работ – защита пяти отчётов;

— приобретения навыков программирования – проведение трёх контрольных работ.

Курс 1, семестр 1 Контроль обучения – Зачет.

Максимальный семестровый рейтинг – **100 баллов.**

Таблица 11.1 Балльные оценки для элементов контроля.

Элементы учебной деятельности	Максимальный балл на 1-ую КТ с начала семестра	Максимальный балл за период между 1КТ и 2КТ	Максимальный балл за период между 2КТ и на конец семестра	Всего за семестр
Посещение занятий	3	3	3	**9**
Тестовый контроль	4	4	4	**12**
Контрольные работы на практических занятиях	13	14	14	**41**
Лабораторные работы	9	10	10	**29**
Компонент своевременности	3	3	3	**9**
Итого максимум за период:	**32**	**34**	**34**	**100**
Нарастающим итогом	**32**	**66**	**100**	

Студент, набравший менее 60 баллов в течение семестра, считается неуспевающим. Студент, выполнивший все запланированные лабораторные работы и набравший сумму 60 и более баллов, получает зачёт «автоматом».

Таблица 11.2 Пересчет баллов в оценки за контрольные точки

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

Курс 1, семестр 2 Контроль обучения – экзамен.

Максимальный семестровый рейтинг – **100 баллов.**

Таблица 11.3 Балльные оценки для элементов контроля.

Элементы учебной деятельности	Максимальный балл на 1-ую КТ с начала семестра	Максимальный балл за период между 1КТ и 2КТ	Максимальный балл за период между 2КТ и на конец семестра	Всего за семестр
Посещение занятий	3	3	3	**9**
Тестовый контроль	4	4	4	**12**
Контрольные работы на практических занятиях	4	4	4	**12**
Лабораторные работы	8	10	10	**28**
Компонент своевременности	3	3	3	**9**
Итого максимум за период:	**22**	**24**	**24**	**70**
Сдача экзамена (максимум)				**30**
Нарастающим итогом	**22**	**46**	**70**	**100**

Итоговый экзамен по дисциплине обязателен. Студент, набравший менее 50 баллов в течение семестра, к экзамену не допускается. Он должен сдать все не зачтённые лабораторные и контрольные работы.

Согласована на портале № 5770

Итоговый экзамен по дисциплине включает решение задачи (написание программы) и ответы на два теоретических вопроса билета. На экзамене студент может набрать до 12 баллов за решение задачи и до 9 баллов за ответ на каждый теоретический вопрос.

Студент, набравший в течение семестра более 65 баллов, может не решать экзаменационную задачу.

Студент, не решивший задачу, к теоретической части экзамена не допускается. Не сдавшим экзамен считается не решивший задачу или набравший менее двенадцати баллов в сумме.

Студент, не сдавший экзамен, обязан его пересдать в установленном в ТУСУРе порядке. При неудовлетворительной сдаче экзамена или при неявке на экзамен экзаменационная составляющая рейтинга приравнивается к нулю.

Таблица 11.4 Пересчёт баллов в оценки за контрольные точки

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

Таблица 11.5 – Пересчет суммы баллов в традиционную и международную оценку

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

12. Учебно-методическое и информационное обеспечение дисциплины:

12.1 Основная литература

1. C/C++. Программирование на языке высокого уровня : учебник для вузов / Т. А. Павловская. - СПб. : ПИТЕР, 2013. - 461 с. (35 экз.)

2. Языки и методы программирования : учебник для вузов / И. Г. Головин, И. А. Волкова. - М. : Академия, 2012. - 304 с. (30 экз.)

12.2 Дополнительная литература

1. Технологии программирования [Электронный ресурс] : учебное пособие / В. В. Кручинин ; Томский государственный университет систем управления и радиоэлектроники (Томск). - Электрон. текстовые дан. - Томск : [б. и.], 2013. – Режим доступа : http://edu.tusur.ru/training/publications/2834 (дата обращения 13.05.2-16г)

2. Язык программирования C++. Лекции и упражнения [Текст] : научно-популярное издание / С. Прата ; пер.: Ю. И. Корниенко, А. А. Моргунова. - 6-е изд. - М. : Вильямс, 2013. - 1248 с (15 экз.)

3. Подбельский В.В., Фомин С.С. Программирование на языке Си : Учебное пособие для ВУЗов. – 2-е изд., доп. – М.: Финансы и статистика, 2007. – 600 с.

12.3 Перечень методических указаний

По лабораторным работам

1. Программирование на языках высокого уровня: Методические указания по выполнению лабораторных работ и самостоятельной работе для студентов 230200 «Информационные системы» / Панасенко Е. А. – 2012. 12 с - 12 с. - Режим доступа: https://edu.tusur.ru/training/publications/2198

По практическим занятиям

2. Программирование и основы алгоритмизации [Электронный ресурс] : лабораторный практикум / Е. В. Истигечева, А. В. Мельников ; Томский государственный университет систем управления и радиоэлектроники - Томск : 2015. - 31. - Режим доступа: https://edu.tusur.ru/training/publications/5024

По самостоятельной работе

3. Программирование и основы алгоритмизации [Электронный ресурс] : методические указания по самостоятельной работе / Е. В. Истигечева, А. В. Мельников ; Томский государственный университет систем управления и радиоэлектроники - Томск : 2015. - 11 с. - Режим доступа: https://edu.tusur.ru/training/publications/5023

Согласована на портале № 5770

12.4 Лицензионное и свободно распространяемое программное обеспечение

Операционная система MS Windows XP, пакет Open Office, CodeBlock, Visual C Express Edition.

12.5 Информационно-справочные и поисковые системы

Информационно-справочные и поисковые системы сети Интернет.

13. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Для проведения практический занятий и лабораторных работ по дисциплине используются персональный ПК с процессором Pentium 4, Операционная система MS Windows XP, свободно распространяемые пакеты Open Office, CodeBlock, Visual C Express Edition

Согласована на портале № 5770

Приложение к рабочей программе

Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования

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

У Т В Е Р Ж Д А Ю
Проректор по учебной работе
_____ П.Е. Троян
«28» 09 2016 г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ПО УЧЕБНОЙ ДИСЦИПЛИНЕ

ПРОГРАММИРОВАНИЕ

Уровень основной образовательной программы бакалавриат

Направление подготовки 09.03.01 – Информатика и вычислительная техника

Профиль Программное обеспечение средств вычислительной техники и автоматизированных систем

Форма обучения очная

Факультет систем управления

Кафедра автоматизированных систем управления

Курс 1

Семестр 1, 2

Учебные планы наборов 2013, 2014, 2015, 2016 тодов и последующих лет

Зачет 1 семестр

Экзамен 2 семестр

Томск 2016

Согласована на портале № 5770

1 ВВЕДЕНИЕ

Фонд оценочных средств (ФОС) является приложением к рабочей программе дисциплины «**Программирование**» и представляет собой совокупность контрольно-измерительных материалов (типовые задачи (задания), контрольные работы, тесты и др.) и методов их использования, предназначенных для измерения уровня достижения студентом установленных результатов обучения.

ФОС по дисциплине (практике) используется при проведении текущего контроля успеваемости и промежуточной аттестации студентов. Перечень закрепленных за дисциплиной «**Программирование**» компетенций приведен в таблице 1.

Таблица 1 – Перечень закрепленных за дисциплиной компетенций

Код	Формулировка компетенции	Этапы формирования компетенции
ОПК-2	Способность осваивать методики использования программных средств для решения практических задач.	***Знать:*** методы структурного программирования, понятие данных и их структуры; ***Уметь:*** разрабатывать и отлаживать эффективные алгоритмы и прикладное ПО с использованием современных технологий программирования; ***Владеть:*** навыками разработки и отладки программ в современных средах.
ПК-1	Способность разрабатывать модели компонентов информационных систем, включая модели баз данных и модели интерфейсов «человек – ЭВМ».	***Знать:*** методы разработки программных приложений. ***Уметь:*** программировать приложения; ***Владеть:*** навыками разработки программных приложений в современных средах.

2 РЕАЛИЗАЦИЯ КОМПЕТЕНЦИЙ

2.1 Компетенция ОПК-2

ОПК-2: Способность осваивать методики использования программных средств для решения практических задач.

Для формирования компетенции необходимо осуществить ряд этапов. Этапы формирования компетенции, применяемые для этого виды занятий и используемые средства оценивания, представлены в таблице 2.

Таблица 2 – Этапы формирования компетенции и используемые средства оценивания

Состав	Знать	Уметь	Владеть
Содержание этапов	Методы структурного программирования, понятие данных и их структуры, этапы разработки ПО.	Разрабатывать и отлаживать эффективные алгоритмы и прикладное ПО с использованием современных технологий программирования.	Навыками разработки и отладки ПО в современных средах.
Виды занятий	Лекции. Практические занятия. Самостоятельная работа.	Практические занятия. Лабораторные работы. Самостоятельная работа.	Практические занятия. Индивидуальные задания. Самостоятельная работа.
Используемые средства оценивания	Контрольная работа. Устный опрос. Контроль выполнения домашнего задания. Экзамен.	Проверка правильности выполнения заданий. Контрольная работа. Отчет о лаб. работе. Экзамен.	Проверка правильности выполнения заданий. Проверка индивидуального задания. Отчет о лаб. работе.

Общие характеристики показателей и критериев оценивания компетенции на всех этапах приведены в таблице 3.

Таблица 3 – Общие характеристики показателей и критериев оценивания компетенции по этапам.

Показатели и критерии	Знать	Уметь	Владеть
ОТЛИЧНО (высокий уровень)	Обладает фактическими и теоретическими знаниями в пределах изучаемой области с пониманием границ применимости.	Обладает диапазоном практических умений, требуемых для развития творческих решений, абстрагирования проблем.	Контролирует работу, проводит оценку, совершенствует действия работы.
ХОРОШО (базовый уровень)	Знает факты, принципы, процессы, общие понятия в пределах изучаемой области.	Обладает диапазоном практических умений, требуемых для решения опреде-	Берет ответственность за завершение задач в исследовании, приспосабливает свое поведение

Согласована на портале № 5770

		ленных проблем в области исследования.	к обстоятельствам в решении проблем.
УДОВЛЕТВОРИТЕЛЬНО (низкий уровень)	Обладает низким уровнем общих знаний.	Обладает умениями на низком уровне, которые не достаточны для выполнения даже простых задач.	Работает только при прямом наблюдении.

Формулировка показателей и критериев оценивания данной компетенции приведена в таблице 4.

Таблица 4 – Показатели и критерии оценивания компетенции на этапах

Показатели и критерии	**Знать**	**Уметь**	**Владеть**
ОТЛИЧНО (высокий уровень)	Знает все методы структурного программирования, все типы данных и их структуры, динамические и статические структуры данных, этапы разработки ПО.	Умеет разрабатывать и отлаживать эффективные алгоритмы и прикладное ПО любого уровня сложности с использованием современных технологий программирования.	Свободно владеет навыками разработки и отладки ПО в современных средах.
ХОРОШО (базовый уровень)	Знает все методы структурного программирования, основные типы данных и их структуры, этапы разработки ПО.	Умеет разрабатывать и отлаживать алгоритмы и прикладное ПО среднего уровня сложности с использованием современных технологий программирования.	Владеет навыками разработки и отладки ПО в современных средах.
УДОВЛЕТВОРИТЕЛЬНО (низкий уровень)	Знает основные методы структурного программирования и типы данных, имеет понятие о структуре данных	Умеет разрабатывать простые алгоритмы и прикладное ПО с использованием современных технологий программирования.	Владеет навыками разработки несложного ПО в одной определенной среде.

2.2 Компетенция ПК-1

ПК-1: Способность разрабатывать модели компонентов информационных систем, включая модели баз данных и модели интерфейсов «человек – ЭВМ».

Для формирования компетенции необходимо осуществить ряд этапов. Этапы формирования компетенции, применяемые для этого виды занятий и используемые средства оценивания, представлены в таблице 5.

Таблица 5 – Этапы формирования компетенции и используемые средства оценивания

Состав	**Знать**	**Уметь**	**Владеть**
Содержание этапов	Все типы данных, статические и динамические структуры данных	Разрабатывать модели компонентов ИС.	Навыками разработки компонентов ИС в современных средах
Виды занятий	Лекции. Практические занятия. Самостоятельная работа.	Практические занятия. Лабораторные работы. Самостоятельная работа.	Практические занятия. Индивидуальные задания. Самостоятельная работа.
Используемые средства оценивания	Контрольная работа. Устный опрос. Контроль выполнения домашнего задания. Экзамен.	Проверка правильности выполнения заданий. Контрольная работа. Отчет о лаб. работе. Экзамен.	Проверка правильности выполнения заданий. Проверка индивидуального задания. Отчет о лаб. работе.

Общие характеристики показателей и критериев оценивания компетенции на всех этапах приведены в таблице 6.

Таблица 6 – Общие характеристики показателей и критериев оценивания компетенции по этапам

Показатели и критерии	**Знать**	**Уметь**	**Владеть**
ОТЛИЧНО (высокий уровень)	Обладает фактическими и теоретическими знаниями в пределах изучаемой области с пониманием границ применимости.	Обладает диапазоном практических умений, требуемых для развития творческих решений, абстрагирования проблем.	Контролирует работу, проводит оценку, совершенствует действия работы.
ХОРОШО	Знает факты, принципы,	Обладает диапазоном практи-	Берет ответственность за завер-

Согласована на портале № 5770

(базовый уровень)	процессы, общие понятия в пределах изучаемой области.	ческих умений, требуемых для решения определенных проблем в области исследования.	шение задач в исследовании, приспосабливает свое поведение к обстоятельствам в решении проблем.
УДОВЛЕТВОРИТЕЛЬНО (низкий уровень)	Обладает низким уровнем общих знаний.	Обладает умениями на низком уровне, которые не достаточны для выполнения даже простых задач.	Работает только при прямом наблюдении.

Формулировка показателей и критериев оценивания данной компетенции приведена в таблице 7.

Таблица 7 – Показатели и критерии оценивания компетенции на этапах

Показатели и критерии	**Знать**	**Уметь**	**Владеть**
ОТЛИЧНО (высокий уровень)	Знает все методы структурного программирования, все типы данных и их структуры, динамические и статические структуры данных, этапы разработки ПО.	Умеет разрабатывать модели компонентов ИС любого уровня сложности с использованием современных технологий программирования.	Свободно владеет навыками разработки моделей компонентов ИС в современных средах.
ХОРОШО (базовый уровень)	Знает все методы структурного программирования, основные типы данных и их структуры, этапы разработки ПО.	Умеет разрабатывать модели компонентов ИС среднего уровня сложности с использованием современных технологий программирования.	Владеет навыками разработки моделей компонентов ИС в современных средах.
УДОВЛЕТВОРИТЕЛЬНО (низкий уровень)	Знает основные методы структурного программирования и типы данных, имеет понятие о структуре данных	Умеет разрабатывать простые модели компонентов ИС с использованием современных технологий программирования.	Владеет навыками разработки несложных моделей компонентов ИС в одной определенной среде.

3 ТИПОВЫЕ КОНТРОЛЬНЫЕ ЗАДАНИЯ

Для реализации вышеперечисленных задач обучения используются следующие материалы: типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы, в составе, приведенном ниже.

3.1 Темы практических занятий

3.1.1 Первый семестр

1. Знакомство с языком Си и средой программирования. Базовые типы данных. Форматированный ввод-вывод. Линейные алгоритмы. Разработка простой программы.
2. Нисходящее проектирование программ. Базовые конструкции структурного программирования. Способы описания алгоритмов.
3. Структурные конструкции в языке Си. Классы памяти, инициализация переменных. Разработка простой нелинейной программы.
4. Рекуррентные последовательности и алгоритмы. Разработка программы вычисления приближённого значения бесконечной суммы.
5. Функции в языке Си. Определение, объявление, прототип. Механизм передачи параметров. Разработка программы с использованием функции.
6. Алгоритмы поиска корней функции (методы дихотомии, секущих, Ньютона, итераций).
7. Передача функции в качестве параметра функции.
8. Проверка правильности алгоритмов. Методы отладки.
9. Массивы в языке Си. Работа с одномерными массивами.
10. Указатели. Объявление, инициализация, операции. Использование указателей.
11. Указатели и массивы.
12. Динамические массивы.
13. Алгоритмы линейного поиска.
14. Алгоритм поиска в одномерном массиве методом дихотомии.

Согласована на портале № 5770

3.1.2 Второй семестр

1. Алгоритмы сортировки массивов (методы простого обмена, простого выбора, простых вставок).
2. «Быстрые» алгоритмы сортировки.
3. Рекурсивные алгоритмы.
4. Обработка символьных строк в языке Си. Массивы строк. Сортировка строк.
5. Потоковый ввод/вывод. Обработка текстовых файлов.
6. Структурные типы данных в языке Си. Объявление типа. Объявление переменной. Инициализация. Указатели на структуры. Доступ к значениям полей. Структуры и функции.
7. Потоковый ввод/вывод. Обработка бинарных файлов.
8. Динамические информационные структуры (линейный список, дерево).
9. Создание простого интерфейса пользователь-компьютер.

3.2 Темы лабораторных работ

3.2.1 Первый семестр

1. Разработка линейных алгоритмов на языке Си.
2. Разработка циклических алгоритмов на языке Си.
3. Разработка программы табулирования функции, заданной рядом Тейлора.
4. Создание функций. Прототипы функций. Механизм передачи параметров.
5. Разработка программы поиска корней функции.
6. Современные методы отладки алгоритмов.
7. Разработка на языке Си программы обработки одномерного массива.
8. Разработка на языке Си программы обработки двумерного массива.

3.2.2 Второй семестр

1. Разработка программы, использующей внешнюю память.
2. Разработка программы упорядочения элементов двумерного числового массива.
3. Разработка рекурсивных алгоритмов.
4. Разработка программы обработки строк.
5. Разработка программы упорядочения строковых данных.
6. Разработка программы обработки структур.
7. Разработка программы на языке Си с использованием динамических линейных списков.
8. Создание программы на языке Си простого интерфейса пользователь-компьютер.

3.3 Примеры вариантов контрольных работ

3.3.1 Контрольная работа № 1.

ВАРИАНТ 1

1. Напишите программу, выводящую ваше имя и адрес.
2. При условии, что переменная **value** имеет тип **int**, определите, какой будет получен в результате выполнения следующего цикла:

```
for ( value = 36; value > 0; value /= 2)
printf("%3d", value);
```

Какие проблемы могли бы возникнуть, если бы переменная **value** имела тип **double** вместо **int**?

3. Найдите все ошибки в следующей программе?

```
#include <stdio.h>
int main (void)
{
 char ch;
 int lc =0; /* счетчик строчных символов
 int lc = 0; /* счетчик прописных символов
 int lc = 0; /* счетчик других символов
 while ( (ch = getchar()) != '#')
 {
  if ( ' a ' <= ch >= ' z ' )
   lc + +;
   else if (' (ch < 'A' ) 11 !(ch> ' Z')
```

Согласована на портале № 5770

```
    uc++;
    oc++;
 }
 printf(%d строчных, %d прописных, %d других, lc, uc, oc);
 return ( );
}
```

3.3.2 Контрольная работа № 2.

ВАРИАНТ 1

1. Ввести с клавиатуры последовательность **n** целых чисел **a1, a2, .., an.** Получить сумму тех чисел, которые удовлетворяют условию **|ai| < i** . Для организации цикла использовать оператор **for**. Задачу решить, не используя массив. К каждой строке программы написать комментарий.

3.3.3 Контрольная работа № 3.

ВАРИАНТ 1

1. Чем отличаются механизмы передачи параметров по ссылке и по значению.
2. Дан массив целых чисел: 50 60 15 90 11 35 5. Отсортировать его вручную методом простых вставок. Показать состояние массива на каждом шаге.
3. Написать программу на языке Си, которая читает бинарный файл вещественных чисел, выводит на экран самый большой по значению компонент файла и его номер.

3.3.4 Контрольная работа № 4.

ВАРИАНТ 1

1. В каком случае использование рекурсивных алгоритмов является оправданным?
2. Написать на Си рекурсивную функцию, решающую следующую задачу.

Даны неотрицательные числа **n, m**. Вычислить значение функции

$$A(n,m)=\begin{cases} m+1; & n=1; \\ A(n-1,1); & n\neq 0, m=0; \\ A\big(n-1,A(n,m-1)\big); & n>0, m>0. \end{cases}$$

3.4 Примеры вариантов индивидуальных заданий на семестр

3.4.1 Первый семестр

ВАРИАНТ 1

1. Робот может перемещаться в четырех направлениях («С» — север, «З» — запад, «Ю» — юг, «В» — восток) и принимать три цифровые команды: 0 — продолжать движение, 1 — поворот налево, −1 — поворот направо. Дан символ C — исходное направление робота и целое число N — посланная ему команда. Вывести направление робота после выполнения полученной команды.
2. Дано вещественное число X ($|X| < 1$) и целое число N (> 0). Найти значение выражения $X - X^3/3 + X^5/5 - \ldots + (-1)^N \cdot X^{2\cdot N+1}/(2\cdot N+1)$. Полученное число является приближенным значением функции arctg в точке X.
3. Дано целое число N и набор из N целых чисел, содержащий по крайней мере два нуля. Вывести сумму чисел из данного набора, расположенных между последними двумя нулями (если последние нули идут подряд, то вывести 0). *Массивы не использовать.*
4. Описать функцию *Even*(K), возвращающую ненулевое значение, если целый параметр K является четным, и ноль в противном случае. С ее помощью найти количество четных чисел в наборе из N псевдослучайных целых чисел.
5. Дан целочисленный массив размера N, содержащий ровно два одинаковых элемента. Найти номера одинаковых элементов и вывести эти номера в порядке возрастания.
6. Дан массив размера N. Заменить каждый элемент массива на среднее арифметическое этого элемента и его ближайших соседей. *Вспомогательные массивы не использовать.*
7. Дано целое число L (> 0) и целочисленный массив размера N. Заменить каждую серию массива, длина которой равна L, на один элемент с нулевым значением. *Серия — группа подряд идущих одинаковых элементов. Длина серии — количество этих элементов (может быть равна 1).*
8. Дана матрица размера M×N. Удалить строку, содержащую минимальный элемент матрицы.
9. Дана квадратная матрица порядка M. Обнулить элементы матрицы, лежащие одновременно ниже главной диагонали (включая эту диагональ) и выше побочной диагонали (также включая эту диагональ). Условный оператор не использовать.

3.4.2 Второй семестр

ВАРИАНТ 1

Согласована на портале № 5770

1. Написать функцию, возвращающую номер строки целочисленной прямоугольной матрицы, в которой находится самая длинная серия одинаковых элементов.
2. Дана строка, изображающая целое положительное число. Вывести сумму цифр этого числа.
3. Даны строки S и S0. Удалить из строки S первую подстроку, совпадающую с S0. Если совпадающих подстрок нет, то вывести строку S без изменений.
4. Дана строка-предложение. Зашифровать ее, поместив вначале все символы, расположенные на четных позициях строки, а затем, в обратном порядке, все символы, расположенные на нечетных позициях (например, строка «Программа» превратится в «ргамамроП»).
5. Написать функцию Smooth2(A, N), выполняющую сглаживание вещественного массива A размера N следующим образом: элемент A1 не изменяется, элемент AK (K = 2, . . ., N) заменяется на полусумму исходных элементов AK−1 и AK. Сглаженный массив остаётся на месте исходного. С помощью этой функции выполнить пятикратное сглаживание данного массива A размера N, выводя результаты каждого сглаживания. (Примечание. K – порядковый номер элемента!!!)
6. Для хранения данных о планшетных сканерах описать структуру вида:

```
struct scan_info{
    char model[25];  // наименование модели
    int price;       // цена
    double x_size;   // горизонтальный размер области сканирования
    double y_size;   // вертикальный размер области сканирования
    int optr;        // оптическое разрешение
    int grey;        // число градаций серого
};
```

Написать функцию, которая записывает в бинарный файл данные о сканере из приведенной структуры. Структура файла: в первых двух байтах размещается значение типа **`int`**, определяющее количество сделанных в файл записей; далее без пропусков размещаются записи о сканерах.
Написать функцию, которая извлекает из этого файла данные о сканере в структуру типа **`scaninfo`**. Обязательный параметр — номер требуемой записи. Функция должна возвращать нулевое значение, если чтение прошло успешно, и **`-1`** в противном случае.
Привести пример программы, создающей файл с данными о сканерах (данные вводятся с клавиатуры) — 6-8 записей и выводящей на дисплей данные о запрошенной записи. Все необходимые данные для функций должны передаваться им в качестве параметров. Использование глобальных переменных в функциях не допускается.

3.5 Темы для самостоятельной работы (темы рефератов)

1. Синтаксис языка Си (тема 1).
2. Препроцессор языка Си (тема 2).
3. «Быстрые» алгоритмы сортировки (тема 6).
4. Связные списки (тема 12).

3.6 Вопросы для подготовки к теоретическому зачету

Зачёт сдаётся студентом, выполнившим не все контрольные работы или индивидуальные задания.

1. Структурное программирование, метод пошаговой детализации. Примеры.
2. Структурное программирование, его базовые конструкции. Примеры.
3. Структурное программирование, принцип сквозного контроля. Примеры.
4. Структура программы на языке Си.
5. Операции в языке Си.
6. Типы данных в языке Си. Примеры объявления и использования.
7. Препроцессорные средства include, define.
8. Переключатель в Си (множественный выбор).
9. Операторы цикла в языке Си.
10. Операторы ветвления в языке Си.
11. Определение функции в языке Си. Вызов функции. Фактические и формальные параметры.
12. Адреса и указатели в языке Си.
13. Операции над указателями.
14. Массивы в языке Си. Примеры описаний и использований.
15. Понятие рекуррентной последовательности.
16. Алгоритм нахождения «бесконечной» суммы.

3.7 Вопросы и примеры задач для подготовки к экзамену

3.7.1 Вопросы:

Группа 1. Алгоритмизация

Согласована на портале № 5770

1. Понятие алгоритма. Примеры алгоритмов. Формы записи алгоритмов.
2. Структурное программирование, его базовые конструкции. Примеры.
3. Понятие рекуррентного алгоритма и рекуррентной последовательности.
4. Алгоритмы вычисления n-го члена и суммы конечного числа элементов рекуррентной последовательности.
5. Вычисление бесконечных сумм на основе рекуррентной последовательности.
6. Задача нахождения корней функции. Метод дихотомии и его алгоритм.
7. Задача нахождения корней функции. Метод Ньютона и его алгоритм.
8. Задача нахождения корней функции. Метод хорд (линейной интерполяции) и его алгоритм.
9. Основные источники ошибок в программе и методы борьбы с ними.
10. Понятие подпрограммы. Разновидности подпрограмм.
11. Понятие массива. Работа с массивами. Примеры задач.
12. Задача информационного поиска, её разновидности. Алгоритмы нахождения минимального элемента и его номера в последовательностях с различными и совпадающими элементами.
13. Задача информационного поиска. Алгоритмы нахождения номера элемента с заданным значением в последовательности из различных элементов.
14. Задача сортировки массивов. Алгоритм простого выбора.
15. Задача сортировки массивов. Алгоритм простого обмена.
16. Задача сортировки массивов. Алгоритм простых вставок.
17. Сортировка массивов. Алгоритм Шелла.
18. Сортировка массивов. Алгоритм Хоара.
19. Упорядочение нечисловых массивов. Лексикографический принцип.

Группа 2. Язык программирования СИ

20. Алгоритмический язык Си. Основные особенности. Алфавит.
21. Объекты программы на Си.
22. Основные операторы языка Си.
23. Ввод-вывод в языке Си. Примеры. Структура программы.
24. Типы данных языка Си.
25. Арифметические константы.
26. Символьные и строковые константы.
27. Переменные и их описания.
28. Арифметические и логические операции.
29. Преобразования типов.
30. Унарные операции в Си.
31. Побитовые операции в Си.
32. Операции и выражения присваивания. Условные выражения.
33. Ранги операций и порядок вычисления выражений.
34. Условные операторы в Си.
35. Оператор-переключатель.
36. Операторы цикла в Си.
37. Операторы передачи управления.
38. Функции в языке Си. Оформление функций.
39. Структура Си-программы.
40. Область действия имени. Классы памяти.
41. Основные правила инициализации.
42. Адресная арифметика.
43. Данные типа указатель и работа с ними.
44. Указатели и функции.
45. Указатели и массивы. Массивы указателей.
46. Стандартный (бесформатный) ввод-вывод.
47. Форматный ввод-вывод.
48. Работа со строками в Си.
49. Структуры.
50. Работа с файлами.

3.7.2 Примеры задач:

1. Дано натуральное число n и последовательность символов $S1$, …, Sn (строка). Группы символов, разделенные одним или несколькими пробелами, будем называть словами. Подсчитать количество букв a в последнем слове данной строки.
2. По введённому символу установить, в каких позициях его двоичного кода записаны нули.

Согласована на портале № 5770

3. Используя рекуррентный алгоритм вычислить $a^x = \sum_{k=0}^{\infty} \frac{(x \ln a)^k}{k!}$, где a задать с клавиатуры, x=0...10.
4. Дана матрица размера M × N. Найти номера строки и столбца элемента матрицы, наиболее близкого к среднему значению всех ее элементов.
5. Дан текстовый файл `F`. Переписать его в файл `G`, заменив каждый символ `'0'` символом `'1'`.
6. Определить структуру для хранения анкетных данных студентов: **`{Фамилия, Имя, Отчество, Номер группы}`.** Ввести данные для нескольких студентов различных групп и распечатать список студентов заданной группы.

4 МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ

Для обеспечения процесса обучения и решения задач обучения используются следующие материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций, в составе:

1. Учебное пособие по дисциплине «Программирование» приведено в рабочей программе в разделе 12.1 [1,2].
2. Методические указания по лабораторным работам студентов всех форм обучения приведены в рабочей программе в разделе 12.3 [6, 8].
3. Методические указания к практическим занятиям приведены в рабочей программе в разделе 12.3 [6, 7].
4. Методические указания по самостоятельной работе студентов всех форм обучения приведены в рабочей программе в разделе 12.3 [6, 7].

– C/C++. Программирование на языке высокого уровня: учебник для вузов / Т. А. Павловская. - СПб. : ПИТЕР, 2013. - 461 с. (35 экз.)
– Языки и методы программирования : учебник для вузов / И. Г. Головин, И. А. Волкова. - М. : Академия, 2012. - 304 с. (30 экз.)
– Программирование на языках высокого уровня: Методические указания по выполнению лабораторных работ и самостоятельной работе для студентов 230200 «Информационные системы» / Панасенко Е. А. – 2012. 12 с - 12 с. - Режим доступа: https://edu.tusur.ru/training/publications/2198
– Программирование и основы алгоритмизации [Электронный ресурс] : лабораторный практикум / Е. В. Истигечева, А. В. Мельников ; Томский государственный университет систем управления и радиоэлектроники - Томск : 2015. - 31. - Режим доступа: https://edu.tusur.ru/training/publications/5024
– Программирование и основы алгоритмизации [Электронный ресурс] : методические указания по самостоятельной работе / Е. В. Истигечева, А. В. Мельников ; Томский государственный университет систем управления и радиоэлектроники - Томск : 2015. - 11 с. - Режим доступа: https://edu.tusur.ru/training/publications/5023

Дополнительная литература
– Технологии программирования [Электронный ресурс] : учебное пособие / В. В. Кручинин ; Томский государственный университет систем управления и радиоэлектроники (Томск). - Электрон. текстовые дан. - Томск : [б. и.], 2013. – Режим доступа : http://edu.tusur.ru/training/publications/2834 (дата обращения 13.05.2-16г)
– Язык программирования C++. Лекции и упражнения [Текст] : научно-популярное издание / С. Прата ; пер.: Ю. И. Корниенко, А. А. Моргунова. - 6-е изд. - М. : Вильямс, 2013. - 1248 с (15 экз.)
– Программирование на языке Си : Учебное пособие для вузов / В. В. Подбельский, С. С. Фомин. - 2-е изд., доп. - М. : Финансы и статистика, 2007. - 600 с. (2 экз.)
– C++: для начинающих : самоучитель. Пер. с англ. / Г. Шилдт ; пер. К. Г. Финогенов. - М. : ЭКОМ, 2007. – 639 с. (1 экз.)
– Программирование на языке C : пер. с англ. / С. Кочан. - 3-е изд. - М. : Вильямс, 2007. - 489 с. (2экз.)
– Кент, Джефф. C++. Основы программирования : самоучитель / Д. Кент ; пер. Ю. В. Кирпичев. - М. : НТ Пресс, 2010. - 368 с. (1 экз.)

Согласована на портале № 5770