



Факультет	Математики, физики и информатики	
Кафедра	Информатики информационных технологий	
Направление подготовки	020302 Фундаментальная информатика и информационные технологии	
Профиль	Открытые информационные системы	
	Основы программирования	Б1.В.ОД7

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тульский государственный педагогический университет им. Л. Н. Толстого»
ФГБОУ ВПО «ТГПУ им.Л.Н.Толстого»

УТВЕРЖДЕНА

на заседании Ученого совета университета

Протокол № 2

«11» февраля 2016 г.


Учебная программа дисциплины «Основы программирования»

Трудоемкость: 4 зачетные единицы

Квалификация (степень) выпускника: академический бакалавр

Форма обучения: очная

Рассмотрена на заседании кафедры информатики и информационных технологий
протокол № 3 от 18 ноября 2015 г.

Заведующий кафедрой  А.В.Якушин

Одобрена на заседании Ученого совета факультета
математики, физики и информатики
протокол № 5 от «17» декабря 2015 г.

Декан  И.Ю. Реброва

СОДЕРЖАНИЕ

1. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ, СООТНЕСЕННЫХ С ПЛАНИРУЕМЫМИ РЕЗУЛЬТАТАМИ ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ.....	3
2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП.....	3
3. ОБЪЕМ ДИСЦИПЛИНЫ И ВИДЫ УЧЕБНОЙ РАБОТЫ.....	3
4. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ, СТРУКТУРИРОВАННОЕ ПО ТЕМАМ С УКАЗАНИЕМ ОТВЕДЕННОГО НА НИХ КОЛИЧЕСТВА АКАДЕМИЧЕСКИХ ЧАСОВ И ВИДОВ УЧЕБНЫХ ЗАНЯТИЙ.....	4
5. ПЕРЕЧЕНЬ УЧЕБНО-МЕТОДИЧЕСКОГО ОБЕСПЕЧЕНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ «ОСНОВЫ ПРОГРАММИРОВАНИЯ».....	5
6. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ	6
6.1. Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы.....	6
6.2. Описание показателей, критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания	6
6.3. Типовые контрольные задания и иные материалы, характеризующие этапы формирования компетенций в процессе освоения образовательной программы.....	7
6.4. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и/или опыта деятельности, характеризующие этапы формирования компетенций.	9
7. ПЕРЕЧЕНЬ ОСНОВНОЙ И ДОПОЛНИТЕЛЬНОЙ УЧЕБНОЙ ЛИТЕРАТУРЫ, НЕОБХОДИМОЙ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ	10
7.1 Основная литература:	10
7.2 Дополнительная литература:	10
8. ПЕРЕЧЕНЬ РЕСУРСОВ ИНФОРМАЦИОННО-ТЕЛЕКОММУНИКАЦИОННОЙ СЕТИ «ИНТЕРНЕТ», НЕОБХОДИМЫХ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ	10
9. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ.	11
10. ПЕРЕЧЕНЬ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, ИСПОЛЪЗУЕМЫХ ПРИ ОСУЩЕСТВЛЕНИИ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ, ВКЛЮЧАЯ ПЕРЕЧЕНЬ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ И ИНФОРМАЦИОННЫХ СПРАВОЧНЫХ СИСТЕМ.....	11
11. ОПИСАНИЕ МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЙ БАЗЫ, НЕОБХОДИМОЙ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ «ОСНОВЫ ПРОГРАММИРОВАНИЯ».....	12
12. АННОТАЦИЯ РАБОЧЕЙ ПРОГРАММЫ ДИСЦИПЛИНЫ «ОСНОВЫ ПРОГРАММИРОВАНИЯ».....	13
13. ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ К РАБОЧЕЙ ПРОГРАММЕ ДИСЦИПЛИНЫ «ОСНОВЫ ПРОГРАММИРОВАНИЯ».....	13

1. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ, СООТНЕСЕННЫХ С ПЛАНИРУЕМЫМИ РЕЗУЛЬТАТАМИ ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Достижение планируемых результатов обучения, соотнесенных с общими целями и задачами ОПОП, является целью освоения дисциплины (модуля).

Планируемые результаты освоения образовательной программы (код и название компетенции)	Планируемые результаты обучения	Этапы формирования компетенции в процессе освоения образовательной программы
Способность к разработке требований и проектированию программного обеспечения (ДПК-2)	<p>Выпускник знает:</p> <ul style="list-style-type: none"> • Различные способы формирования массива; • Подходы к использованию указателей; <p>Умеет:</p> <ul style="list-style-type: none"> • использовать в программах файлы в качестве источника и приемника данных; • применять базовые алгоритмы сортировки и поиска при решении конкретных задач; • выполнять операции обработки данных строкового типа. <p>Владеет и (или) имеет опыт деятельности :</p> <ul style="list-style-type: none"> • навыками организации модульного строения программ; обработки многомерных массивов; • использования динамической памяти. 	<p>1 этап из 3 (3 семестр)</p>

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

Дисциплина «Основы программирования» относится к обязательным дисциплинам вариативной части дисциплин направления.

К началу изучения дисциплины студенты по результатам изучения предметов «Основы алгоритмизации» и «Введение в программирование» должны владеть:

- знаниями определения и свойства модели, алгоритма и программы;
- умениями записи алгоритмов и реализации основных алгоритмических конструкций следования, ветвления и повторения в любом языке программирования;
- навыками и опытом деятельности поиска информации и реферирования профессиональной литературы.

Дисциплина «Основы программирования» является базовой для качественного изучения основ и методов объектно-ориентированного программирования, основ и принципов объектно-ориентированного и логического программирования, выполнения выпускной квалификационной работы.

3. ОБЪЕМ ДИСЦИПЛИНЫ И ВИДЫ УЧЕБНОЙ РАБОТЫ

Вид учебной работы	Объем часов/зачет-
г. Тула	
Стр. 3 из 14	

	ных единиц по формам обучения
	очная
Максимальная учебная нагрузка (всего)	144
Контактная работа обучающихся с преподавателем (всего)	44
в том числе:	
лекции с применением мультимедийных технологий и раздаточным материалом для студентов	16
лабораторные занятия с использованием современных информационных технологий по разработке алгоритмов и программ	26
контрольные работы	2
Самостоятельная работа студента (всего)	64
в том числе:	
внеаудиторная самостоятельная работа при подготовке к лабораторным и лекционным занятиям	24
Чтение профессиональной и учебной литературы	10
подготовка к контрольной работе	14
Выполнение заданий для самостоятельной работы в модульной объектно-ориентированной динамической учебной среде Moodle	16
Подготовка к экзамену	36
<i>Промежуточная аттестация в форме: экзамена</i>	

4. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ, СТРУКТУРИРОВАННОЕ ПО ТЕМАМ С УКАЗАНИЕМ ОТВЕДЕННОГО НА НИХ КОЛИЧЕСТВА АКАДЕМИЧЕСКИХ ЧАСОВ И ВИДОВ УЧЕБНЫХ ЗАНЯТИЙ

Наименование тем (разделов).	Количество академических или астрономических часов по видам учебных занятий			
	Занятия лекционного типа	Занятия лабораторного типа	Другие виды учебных занятий	Самостоятельная работа обучающихся
Введение	4			6
Обработка исключений	2	2		8
Введение в библиотеку STL	4	4		10
Контейнеры и итераторы	4	14		14
Функциональные объекты	2	6		16
Контрольная работа			2	
Групповые консультации				10
Подготовка к экзамену				36
ИТОГО	16	26	2	100

Тема 1.1. Введение. Этапы решения задач на ЭВМ. Модель. Алгоритм. Способы записи алгоритма.

Тема 1.2. Обработка исключений. Исключения. Перехват событий. Работа с ошибками во время исполнения программы. Трассировка. Отслеживание и контроль ошибок компилятором. Контроль ошибок со стороны программиста. Назначение и принципы работы с заголовочными файлами. Работа на удаленном рабочем столе. Принципы работы в среде программирования. Средства и методы создания и редактирования решений. Средства отладки программ.

Тема 1.3. Введение в библиотеку STL. Работа с данными с помощью абстрактных библиотек. Работа с вложенными типами данных. Работа со строками. Фактические и формальные параметры функций пользователя. Области действия и видимости переменных. Библиотека функций. Библиотека функций пользователя. Разработка собственных типов данных. Перегрузка типов данных.

Тема 1.4. Контейнеры и итераторы. Обработка сложных типов данных. Инстансы. Экземпляры. Объекты. Перебор инстансов. Сборка мусора. Операторы break, continue, return.

Тема 1.5. Функциональные объекты. Работа с объектами в языке Си. Вложенные функции. Процедурный и объектно-ориентированный подход в работе с итераторами.

5. ПЕРЕЧЕНЬ УЧЕБНО-МЕТОДИЧЕСКОГО ОБЕСПЕЧЕНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ «ОСНОВЫ ПРОГРАММИРОВАНИЯ»

- 1) Методическая система, используемая авторами данной рабочей программы, базируется на оптимальном сочетании активных форм и методов организации учебной деятельности студентов (лекция, беседа, анализ, синтез, мозговой штурм и т.п.), приемов групповой (выполнение и защита заданий по разработке алгоритмов и программ) и самостоятельной работы (разработка и защита программ).
- 2) Все студенты являются активными пользователями ресурса системы LMS MOODLE, поскольку в нем представлены конспекты всех лекций и методические разработки к проведению каждого лабораторного занятия.
- 3) В течение всего периода обучения организуется регулярная проверка и учет выполнения домашних заданий, размещенных в LMS MOODLE.
- 4) Промежуточная аттестация принимается в форме зачета, представляющего собой индивидуальную беседу преподавателя и студента по заранее определенному перечню вопросов с обязательным самостоятельным написанием кодов базовых алгоритмов и функций, разобранных и протестированных на занятиях любого вида в течение семестра.
- 5) Ресурс LMS MOODLE поддерживается в актуальном состоянии.
- 6) Одной из важнейших задач методического обеспечения учебной деятельности студента является формирование устойчивого навыка работы в среде программирования посредством использования ресурсов удаленного рабочего стола, расположенного на сервере кафедры, а также контроль умений студентов читать, анализировать и использовать в работе учебную и профессиональную литературу по программированию.
- 7) По дисциплине используется рейтинг.

6. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ

6.1. Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы

Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы представлен в таблице пункта 1 рабочей программы.

Формирование компетенции «Способность к разработке требований и проектированию программного обеспечения» (ДПК-2) осуществляется в течение трех этапов освоения основной профессиональной образовательной программы. Первый этап формирования компетенции осуществляется в процессе освоения дисциплины «Основы программирования» и одной из дисциплин по выбору «Технологии веб-программирования» или «Математические основы исследований». Второй этап формирования компетенции осуществляется в процессе освоения одной из дисциплин по выбору «Объектно-ориентированное программирование» или «Математика в банковской сфере» и одной из дисциплин по выбору «Офисное программирование» или «Оптимизация в управлении». Третий этап формирования компетенции осуществляется в процессе освоения дисциплины «Компьютерная графика».

6.2. Описание показателей, критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания

Дескриптор компетенций	Показатели оценивания	Критерии оценивания
Знания	различных способов формирования массива; подходов к использованию указателей.	Отметка «5» выставляется, если студент в целом за семестр набрал от 81 до 100 баллов (с учетом баллов, набранных на промежуточной аттестации (экзамене)).
Умения	использовать в программах файлы в качестве источника и приемника данных; применять базовые алгоритмы сортировки и поиска при решении конкретных задач; выполнять операции обработки данных строкового типа.	Отметка «4» выставляется, если студент в целом за семестр набрал от 61 до 80 баллов (с учетом баллов, набранных на промежуточной аттестации (экзамене)).
Навыки и опыт деятельности	навыками организации модульного строения программ; обработки многомерных массивов; использования динамической памяти.	Отметка «3» выставляется, если студент в целом за семестр набрал от 41 до 60 баллов (с учетом баллов, набранных на промежуточной аттестации (экзамене)).

Критерии оценивания компетенций формируются на основе балльно-рейтинговой системы с помощью всего комплекса методических материалов, определяющих процедуры оценивания знаний, умений, навыков и опыта деятельности, характеризующих данный этап формирования компетенций.

6.3. Типовые контрольные задания и иные материалы, характеризующие этапы формирования компетенций в процессе освоения образовательной программы

Задания, направленные на формирование подходов программирования с использованием объектно-ориентированного подхода.

1. Назовите принципы объектно-ориентированного представления программных систем
2. Какие функции есть у любого класса?
3. Что такое деструктор?
4. Какое значение должен возвращать деструктор?
5. Какая функция выполняет начальную инициализацию данных в классе?
6. Что такое класс?
7. Преобразование десятичного числа в шестнадцатеричное.
8. Составьте математическую модель задачи о распознавания простого числа.
9. Найти в последовательности чисел первое число больше A

Вопросы к экзамену

- 1 Язык программирования Си. Общая характеристика языка: место в классификации языков, синтаксис и семантика языка, основные объекты языка, операторы, структура программы.
- 2 Данные числовых типов в языке Си: объявление, характеристика, допустимые операции, приведение типов. Пример использования.
- 3 Операции языка Си. Приоритет операций. Оператор и операция присваивания в языке Си. Множественное присваивание. Выражения.
- 4 Функции форматного ввода и вывода. Параметры. Управляющая строка. Спецификаторы формата. Управляющие символы.
- 5 Алгоритмическая конструкция ветвления: полная и неполная форма, блок-схемы. Условная операция. Условный оператор в языке Си: структура оператора, полная и неполная формы, использование сложных условий. Пример на языке Си.
- 6 Алгоритмическая конструкция выбора: понятие, блок-схема. Оператор выбора в языке Си: структура оператора. Пример программы на Си.
- 7 Циклические алгоритмы: понятие, виды (перечислить). Алгоритмическая конструкция цикла с предусловием (понятие, использование, блок-схема). Оператор цикла for: структура оператора, пример использования.
- 8 Циклические алгоритмы: понятие, виды (перечислить). Алгоритмическая конструкция цикла с предусловием (понятие, использование, блок-схема). Оператор цикла с предусловием while в языке Си: структура оператора, допустимые и недопустимые условия, пример использования.
- 9 Циклические алгоритмы: понятие, виды (перечислить). Алгоритмическая конструкция цикла с постусловием (понятие, использование, блок-схема). Оператор цикла с постусловием в языке Си: структура оператора, допустимые и недопустимые условия, пример использования.
- 10 Типовые циклические алгоритмы: максимум/минимум, сумма/произведение, количество.
- 11 Линейный массив: понятие массива, объявление, инициализация массива, индексация элементов. Формирование и вывод массива.
- 12 Типовые алгоритмы для работы с линейными массивами.
- 13 Задача сортировки массива. Алгоритм сортировки линейного массива методом "пузырька".
- 14 Двумерный массив. Объявление, инициализация двумерного массива, индексация элементов. Формирование и вывод двумерного массива.
- 15 Типовые алгоритмы для обработки двумерного массива (целиком).

- 16 Частичная обработка двумерного массива (по строкам или по столбцам). Типовые алгоритмы.
- 17 Указатели. Понятие указателя, объявление, инициализация. Операции, применимые к указателям. Указатель на указатель.
- 18 Использование указателей при работе с массивами и матрицами. Получение адресов и значений элементов. Последовательный перебор элементов. Примеры.
- 19 Функции в языке Си: понятие, объявление, прототипы функций. Вызов функции. Типы возвращаемых значений.
- 20 Функции в языке Си: параметры формальные и фактические, механизм передачи параметров. Передача параметров «по значению» и «по ссылке». Пример использования.
- 21 Локальные и глобальные переменные. Области действия и области видимости. Экранирование переменных.
- 22 Библиотечные функции. Заголовочные файлы. Подключение библиотек. Функции математической библиотеки.
- 23 Рекурсивное описание функций: база рекурсии, рекурсивный вызов, использование стека. Пример использования.
- 24 Динамическая память: выделение и освобождение памяти, размещение данных в динамической памяти. Выделение и освобождение памяти при работе с одиночными переменными и с массивами, изменение размера массива.
- 25 Динамическая память: выделение и освобождение памяти, размещение данных в динамической памяти. Выделение и освобождение памяти при работе с матрицами. Особенности обработки матриц при работе с динамической памятью.
- 26 Массив символов и строка в языке Си. Ввод и вывод строк. Простейшие алгоритмы сканирования и обработки строки. Пример.
- 27 Строка в языке Си: библиотечные функции для обработки строк. Примеры использования.

Примеры типовых заданий для написания программ

Студенту требуется выбрать задание в зависимости от уровня сложности.

Уровень D

1. Функция **add** для суммирования одномерных массивов типа **int** и **float**.
2. Функции **input** и **output** для ввода и вывода на экран массивов и матриц.
3. Функция, которая меняет местами минимальный и максимальный элементы массива.
4. Функция, возвращающая минимальный из трех передаваемых в нее параметров любого (но одинакового) типа.
5. Функция, находящая минимальный элемент в массиве и заменяющая его заданным числом.
6. Функция, меняющей местами значения двух переданных в нее параметров.
7. Функция, подсчитывающая сумму положительных элементов в массиве.

Уровень C

1. Функция **min_el** поиска минимального элемента в массивах и матрицах различного типа.
2. Функция **null_count** определения количества нулей в массивах и матрицах различного типа.
3. Функция **movement** определения пройденного расстояния при равномерном и равноускоренном движениях тела.
4. Функция **distance** вычисления расстояния между двумя точками в двумерном и трехмерном пространстве.
5. Функция, вычисляющая произведение отрицательных элементов массива.
6. Функция, находящая среднее арифметическое элементов массива.

Уровень В

1. Функция **area**, вычисляющая площадь круга по его радиусу, прямоугольника и треугольника по их сторонам.
2. Функция **perimetr**, вычисляющая периметр круга по его радиусу, прямоугольника и треугольника по их сторонам.
3. Функция, формирующая массив, каждый элемент которого равен минимальному из соответствующих значений двух других массивов – параметров.
4. Функция сортировки элементов массива методом пузырька.
5. Функции: **generate** заполнения массивов и матриц случайными числами заданного диапазона и **output** вывода их на экран.
6. Функция, которая считает сумму элементов главной диагонали матрицы.
7. Функция **triangle** вычисления площади треугольников: равностороннего (по стороне), равнобедренного (по основанию и высоте), обычного (по трем сторонам).
8. Функция **volume** определения объемов: шара по его радиусу ($V=4/3pR^2$), конуса по радиусу основания и высоте ($V=1/3pr^2h$), усеченного конуса по радиусам двух оснований и высоте ($V=1/3ph(r^2+r_1^2+rr_1)$).

Уровень А

1. Функция, формирующая новый массив, упорядоченный по неубыванию, из двух массивов-параметров, уже упорядоченных по неубыванию.
2. Функция **surface_area** определения площади поверхности: шара по его радиусу ($S=4pR^2$), цилиндра по радиусу основания и высоте ($S=2pr(r+h)$), усеченного конуса по радиусам двух оснований и образующей ($S=p((r+r_1)l+r^2+r_1^2)$).
3. Функция, находящая среднее квадратическое значение элементов матрицы.
4. Функция **mult** умножения массива и матрицы на число.
5. Функция сложения двух матриц.
6. Функция **sub_array** нахождения разности двух массивов и матриц.
7. Функция, подсчитывающая количество строк матрицы, сумма элементов которых больше некоторого заданного числа (это число имеет тот же тип, что и элементы матрицы).

6.4. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и/или опыта деятельности, характеризующие этапы формирования компетенций

Рейтинг по дисциплине «Основы программирования»

Составляющие итоговой оценки за дисциплину:

1) Текущий контроль (общий вес 60 баллов):

до 21 балла – посещение занятий;

до 26 баллов – выполнение заданий в ходе выполнения лабораторных работ в LMS Moodle и заданий для самостоятельной работы

до 10 баллов – выполнение отдельно выделенных в методических указаниях к выполнению работ задач повышенной сложности

до 3 баллов – выполнение контрольной работы

2) Итоговый контроль заключается в проведении экзамена (общий вес - 40 баллов). Экзамен проводится по вопросам с обязательным решением задач. Как правило, студент получает два вопроса из приведенного выше списка и одну задачу, готовится в присутствии преподавателя и дает подробные комментарии. Студент, пропускавший занятия в ходе семестра, получает дополнительные вопросы и задачи по каждой пропущенной им теме (на усмотрение преподавателя).

№ п/п	Содержание занятия	количество часов	баллы
1.	Введение	10	7
2.	Обработка исключений	12	10
3.	Введение в библиотеку STL	18	15
4.	Контейнеры и итераторы	32	10
5.	Функциональные объекты	24	15
6.	Контрольная работа	2	3
7.	Групповые консультации	10	
8.	Экзамен	36	40
ИТОГО		144	100

7. ПЕРЕЧЕНЬ ОСНОВНОЙ И ДОПОЛНИТЕЛЬНОЙ УЧЕБНОЙ ЛИТЕРАТУРЫ, НЕОБХОДИМОЙ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ

7.1 Основная литература:

1. Александров, Э.Э. Программирование на языке С в Microsoft Visual Studio 2010: учебное пособие / Э.Э. Александров, В.В. Афонин ; Национальный Открытый Университет "ИНТУИТ". - М. : Интернет-Университет Информационных Технологий, 2010. - 500 с. : ил.; То же [Электронный ресурс]. - URL: //biblioclub.ru/index.php?page=book&id=233564
2. Седжвик, Р. Алгоритмы на С++ / Р. Седжвик. - 2-е изд., испр. - М. : Национальный Открытый Университет «ИНТУИТ», 2016. - 1773 с. : ил. - Библиогр. в кн. ; То же [Электронный ресурс]. - URL: //biblioclub.ru/index.php?page=book&id=429164

7.2 Дополнительная литература:

1. Мейерс, С. Эффективный и современный С++ [Текст] / С. Мейерс – М.: Вильямс, 2016. - 304 с.
2. Прата, С. Язык программирования С. Лекции и упражнения[Текст] / С. Прата – М.: Инфра-М, 2015. - 928 с.
3. Страуструп, Б. Язык программирования С++ [Текст] / Б. Страуструп, – М.:Бином,2015. – 1136с.

8. ПЕРЕЧЕНЬ РЕСУРСОВ ИНФОРМАЦИОННО-ТЕЛЕКОММУНИКАЦИОННОЙ СЕТИ «ИНТЕРНЕТ», НЕОБХОДИМЫХ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ

1. www.intuit.ru
2. <http://microsoft.com/>
3. <http://moodle.tsput.ru/course/view.php?id=4188>

Для качественной организации занятий в соответствии с разработанным курсом необходимо иметь возможность:

- 1) чтения лекций, в том числе с использованием мультимедийных средств обучения;
- 2) работы в компьютерном классе. На компьютерах должно быть установлено программное обеспечение:
 - a. интегрированная среда программирования;
 - b. текстовый редактор Microsoft Word

3) выхода студентов в образовательную систему MOODLE

9. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Дисциплина «Основы программирования» направлена на формирование у студентов знаний, умений и владений в области программирования на одном из языков процедурной парадигмы. В результате изучения данной дисциплины студенты должны научиться решать задачи с помощью ЭВМ, раскладывать сложные задачи на независимые подзадачи, представлять, как можно использовать возможности компьютера для решения прикладных задач. Преподавателю необходимо провести систематизацию и выравнивание знаний студентов в области программирования, поскольку они могут сильно варьироваться вследствие того, что часть студентов обучалась по базовому, а часть – по профильному курсу предмета «Информатика и ИКТ» в среднем звене школы.

Преподавание дисциплины должно включать в себя следующие образовательные технологии:

- 1) Организация лекций с использованием при необходимости мультимедийных технологий;
- 2) Использование в ходе лабораторных работ дидактических материалов в виде: опорных конспектов по теоретической составляющей занятий, файлов с примерами программ и т.п.
- 3) Использование ресурсов LMS MOODLE с целью организации процесса систематизации, приобретения и контроля знаний;
- 4) Организация выполнения индивидуальных заданий с целью усиления приобретения навыков самостоятельного построения моделей, разработки алгоритмов и написания программ;
- 5) Организация презентации индивидуальных заданий с целью формирования навыка владения профессиональной терминологией, грамотного выстраивания цепочки высказываний для доказательства обоснованности построенных моделей предметной области и правильности отражения реальных свойств объектов;
- 6) Формирование у студентов убежденности в необходимости последовательного освоения следующих этапов в образовательной деятельности:
 - a. ознакомься с содержанием и теоретическими основами изучаемой темы;
 - b. рассмотри, обсуди с другом и протестируй задачу, решенную кем-то;
 - c. реши самостоятельно задачу, подобную рассмотренной ранее;
 - d. реши самостоятельно задачу по изучаемой теме.

10. ПЕРЕЧЕНЬ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, ИСПОЛЬЗУЕМЫХ ПРИ ОСУЩЕСТВЛЕНИИ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ, ВКЛЮЧАЯ ПЕРЕЧЕНЬ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ И ИНФОРМАЦИОННЫХ СПРАВОЧНЫХ СИСТЕМ

1. Подписка Microsoft DreamSpark Premium - Сублицензионный договор № S-2042626/M18 от 04.06.2013:
 - 1.1. Средства для разработки и проектирования Visual Studio 2008, 2010, 2012 и 2013 Professional Editions;
 - 1.2. Интегрированная среда разработки Visual Studio Express;
 - 1.3. Операционная система Windows Server 2008 Standard Edition 32-bit;
 - 1.4. Операционная система Windows 8.1 Pro;
 - 1.5. Отдельные программы из Office 2007, Office 2010, Office 2013;

2. Операционная система Microsoft Windows XP Professional Russian – Лицензия № 16698685 от 08.08.2003 г.;
3. Программное обеспечение Microsoft Office XP Professional Win32 Russian– Лицензия № 16698685 от 08.08.2003 г.;
4. Веб-браузеры.
5. Доступ студентов через личные кабинеты к электронным библиотечным системам.
6. Возможность работы студентов на удаленном рабочем столе кафедры информатики и информационных технологий.

**11. ОПИСАНИЕ МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЙ БАЗЫ, НЕОБХОДИМОЙ ДЛЯ
ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ
«ОСНОВЫ ПРОГРАММИРОВАНИЯ»**

Компьютерные аудитории, подключение к локальной сети университета и сети Интернет, права доступа к перечисленному выше программному обеспечению.

12. АННОТАЦИЯ РАБОЧЕЙ ПРОГРАММЫ ДИСЦИПЛИНЫ «ОСНОВЫ ПРОГРАММИРОВАНИЯ»

1. Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения образовательной программы

Компетенция: Способность к разработке требований и проектированию программного обеспечения (ДПК-2).

Выпускник знает:

- Различные способы формирования массива;
- Подходы к использованию указателей;

Умеет:

- использовать в программах файлы в качестве источника и приемника данных;
- применять базовые алгоритмы сортировки и поиска при решении конкретных задач;
- выполнять операции обработки данных строкового типа.

Владеет и (или) имеет опыт деятельности :

- навыками организации модульного строения программ; обработки многомерных массивов;
- использования динамической памяти.

2. Место дисциплины «Основы программирования» в структуре ОПОП

Дисциплина «Основы программирования» относится к обязательным дисциплинам вариативной части дисциплин направления.

К началу изучения дисциплины студенты по результатам изучения предметов «Основы алгоритмизации» и «Введение в программирование» должны владеть:

- знаниями определения и свойства модели, алгоритма и программы;
- умениями записи алгоритмов и реализации основных алгоритмических конструкций следования, ветвления и повторения в любом языке программирования;
- навыками и опытом деятельности поиска информации и реферирования профессиональной литературы.

Дисциплина «Основы программирования» является базовой для качественного изучения основ и методов объектно-ориентированного программирования, основ и принципов объектно-ориентированного и логического программирования, выполнения выпускной квалификационной работы.

3. Объем дисциплины 4 зачетные единицы.

4. Образовательный процесс осуществляется на русском языке.

5. Разработчики:

Клепиков Алексей Константинович, кандидат технических наук, ст. преподаватель кафедры информатики и информационных технологий.

13. ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ К РАБОЧЕЙ ПРОГРАММЕ ДИСЦИПЛИНЫ «ОСНОВЫ ПРОГРАММИРОВАНИЯ»

Изменения к рабочей программе отсутствуют.

Заведующий кафедрой

информатики и информационных технологий



А.В. Якушин,

«24» декабря 2015 г.

Программа составлена в соответствии с требованиями ФГОС ВО.

Разработчики:

Фамилия, имя, отчество	Учёная степень	Учёное звание	Должность	Дата разработки	Подпись
Клепиков Алексей Константинович	кандидат технических наук		ст. преподаватель кафедры информатики и информационных технологий;	24.12.2016	