

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Государственное образовательное учреждение
высшего профессионального образования

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

Кафедра автоматизации обработки информации

Утверждаю:

Зав. каф. АОИ

профессор

_____ Ю.П. Ехлаков

«__» _____ 2016 г.

Информатика (Информатика и программирование)

методические указания к самостоятельной работе
для студентов направления
09.03.04 – «Программная инженерия»

Разработчик:

ст. преподаватель

_____ Н.В. Пермякова

СОДЕРЖАНИЕ

Введение_____	3
1. Общая характеристика самостоятельной работы по дисциплине____	3
2. Структура самостоятельной работы _____	3
3. Темы контрольных работ_____	4
4. Подготовка к лабораторным работам_____	4
5. Темы тестовых опросов_____	5
6. Выполнение домашних заданий_____	5
7. Подготовка к практическим работам_____	6
8. Подготовка к коллоквиуму_____	6
Приложение А. Примерный билет_____	8

ВВЕДЕНИЕ

Дисциплина «Информатика и программирование» относится к базовым дисциплинам профессионального цикла подготовки студентов направления «Программная инженерия» (ПИ).

Целью данного курса является подготовка специалиста, умеющего работать с информацией, хранящейся или обрабатываемой в вычислительных системах, владеющего современными информационными технологиями. Кроме того, дисциплины «Информатика» и «Информатика и программирование» являются базовыми для всех курсов, использующих автоматизированные методы анализа и расчетов, и так или иначе использующих компьютерную технику.

Данные методические указания предназначены для выполнения самостоятельной работы по дисциплине «Информатика и программирование» для студентов, обучающихся по направлениям «Программная инженерия» и «Бизнес-информатика».

1. ОБЩАЯ ХАРАКТЕРИСТИКА САМОСТОЯТЕЛЬНОЙ РАБОТЫ ПО ДИСЦИПЛИНЕ

Самостоятельная работа является важной составляющей в изучении дисциплины и состоит из следующих видов деятельности: проработка лекционного материала для подготовки к тестированию, контрольным работам и коллоквиуму, подготовка к лабораторным работам, выполнение домашних заданий, выполнение командных заданий.

Самостоятельная работа над теоретическим материалом направлена на изучение основных понятий и принципов информатики, изучение языка программирования Си и основ структурного программирования. Самостоятельная работа по подготовке к лабораторным работам направлена на изучение методического материала по лабораторным работам и заготовке шаблонов отчета. Выполнение домашних заданий – полностью самостоятельная работа, направленная на получение навыков самостоятельного составления алгоритмов, программ, реализующих составленные алгоритмы и их дальнейшее тестирование.

Выполнение командного задания подразумевает создание и последующее тестирование программного продукта в команде.

2. СТРУКТУРА САМОСТОЯТЕЛЬНОЙ РАБОТЫ

Объем и виды самостоятельной работы в структуре дисциплины приведены в следующей таблице:

Наименование работы	Кол-во часов	Форма контроля (примеры)
Изучение тем курса, вынесенных на самостоятельное изучение	12	Тестовый опрос, контрольная работа
Подготовка к контрольным работам	33	Контрольная работа
Подготовка к практическим занятиям	6	Защита индивидуальных заданий
Подготовка к лабораторным работам	46	Отчет, защита лабораторной работы
Выполнение домашних заданий	30	Отчет, защита домашних заданий
Подготовка к коллоквиуму	8	Коллоквиум
Подготовка к тестовым опросам	23	Тест
Всего часов самостоятельной работы	158	

3. ТЕМЫ КОНТРОЛЬНЫХ РАБОТ

- № 1. Представление данных в ЭВМ
- № 2. Способы представления алгоритмов
- № 3. Синтаксис языка Си
- № 4. Циклы, проверка условий в языке Си
- № 5. Работа с массивами в языке Си
- № 6. Работа с матрицами в языке Си
- № 7. Работа со строками в языке Си
- № 8. Эффективность алгоритмов
- № 9. Сортировки №1
- №10. Сортировки №2
- № 11. Слияние
- № 12. Поразрядные сортировки
- № 13. Поиск подстроки
- № 14. Бинарные деревья № 1
- №15. Бинарные деревья № 2
- № 16. Объектно-ориентированная методология программирования
- № 17. Разработка простого класса
- № 18. Иерархия классов

4. ПОДГОТОВКА К ЛАБОРАТОРНЫМ РАБОТАМ

Темы лабораторных работ

Ввод-вывод данных. Проверка условий. Геометрия на плоскости

Вычисление суммы бесконечного ряда с заданной точностью.

Статические массивы. Алгоритмы поиска

Динамические матрицы. Функции.

Строки

Текстовые файлы

Двоичные файлы

Односвязные динамические списки.

Управление консолью

Поразрядные сортировки

Бинарный и интерполяционный поиск

Поиск подстроки в строке

Построение BST-деревьев

Операции на BST-деревьях

Знакомство с языком программирования Java. Простые программы

Управляющие конструкции

Массивы и строки

Создание класса. Конструкторы

Внутренние классы

Абстрактные классы и интерфейсы. Наследование и полиморфизм. Создание класса-наследника

Создание класса для работы с файлами

Обработка исключительных ситуаций

Разработка и создание программной мини-системы

При подготовке к лабораторным работам необходимо изучить теоретический материал по заданию к лабораторной работе. Подготовить шаблон отчета и ответы на контрольные вопросы. Для подготовки к занятиям используйте Методические указания к лабораторным работам и практическим занятиям. Электронный вариант указаний доступен по ссылке:

http://aoi.tusur.ru/upload/methodical_materials/MU_LR_PR_Informatika_i_programmirovanie_09_03_04_file_678_697.pdf.

5. ТЕМЫ ТЕСТОВЫХ ОПРОСОВ

Линейные алгоритмы.

Разветвляющиеся алгоритмы.

Циклические алгоритмы.

Функции ввода-вывода.

Рекурсивные функции.

Связные списки.

Тестовые опросы проводятся на лекционных занятиях, о теме теста и времени проведения студенты предупреждаются заранее. Тесты помогают оценить глубину освоения излагаемого на лекционных занятиях материала студентами и подходят для промежуточного контроля.

6. ВЫПОЛНЕНИЕ ДОМАШНИХ ЗАДАНИЙ

Темы домашних заданий:

Системы счисления

Алгоритмизация

Циклические процессы

Файлы

Двусвязные списки

Проверка эффективности алгоритмов

Сортировка подсчетом

Очереди по приоритетам

AVL-деревья

Выполнение домашних заданий подразумевает самостоятельную разработку алгоритма по индивидуальному заданию, представление составленного алгоритма на языке высокого уровня, тестирование написанной программы, демонстрация и защита результатов работы программы.

Для выполнения домашних работ может быть использована следующая дополнительная литература:

1. Кнут Д. Э. Искусство программирования : пер. с англ. / Д. Э. Кнут ; ред. пер. Ю. В. Козаченко ; Станфордский университет. - М. : Вильямс, 2004 - . - (Классический труд. Исправленное и дополненное издание). Т. 3 : Сортировка и поиск : монография / пер. В. Т. Тертышный, пер. И. В. Красиков. - 2-е изд. - М. : Вильямс, 2004. - 822[2] с.

2. Скиена С. Олимпиадные задачи по программированию : Руководство по подготовке к соревнованиям. Пер. с англ. / С. С. Скиена, М. А. Ревилла ; пер. : Б. В. Кучин ; авт. послесл. : В. М. Кирюхин. - М. : КУДИЦ-ОБРАЗ, 2005. - 415[1] с.

3. Вирт Н. Алгоритмы и структуры данных (с примерами на Паскале) : пер. с англ. / Н. Вирт ; пер. Д. Б. Подшивалов. - 2-е изд., испр. . - СПб. : Невский диалект, 2007. - 351[1] с.

4. Павловская Т. А. С/С++. Структурное программирование. Практикум : Учебное пособие для вузов / Т. А. Павловская, Ю. А. Щупак. - СПб. : Питер, 2005. - 238 с.

5. Джехани Н. Программирование на языке Си : Пер. с англ. / Н. Джехани ; пер. И. Г. Шестаков, ред. пер. Б. А. Кузьмин. - М. : Радио и связь, 1988. - 270[2] с.

6. Аляев Ю. А. Алгоритмизация и языки программирования Pascal, C++, Visual Basic : Учебно-справочное пособие / Ю. А. Аляев, О. А. Козлов. - М. : Финансы и статистика, 2004. - 318[2] с.

7. ПОДГОТОВКА К ПРАКТИЧЕСКИМ ЗАНЯТИЯМ

Цикл дисциплины во втором семестре изучения предусматривает 16 часов практических занятий. Во время проведения практических занятий студентам предлагается выполнить 3 индивидуальных задания, темы заданий представлены ниже:

- простые сортировки на месте;
- улучшенные сортировки на месте;
- сортировка слиянием.

Самостоятельная работа при подготовке к практическим занятиям заключается в своевременном изучении теоретического материала по теме занятия и самостоятельной реализации индивидуального задания (разработка алгоритма, программная реализация, тестирование и защита). Теоретический материал по темам практических занятий доступен по ссылке

http://aoi.tusur.ru/upload/methodical_materials/MU_LR_PR_Informatika_i_programmirovanie_09_03_04_file_678_697.pdf.

8. ПОДГОТОВКА К КОЛЛОКВИУМУ

Коллоквиум является заключительным элементом промежуточного контроля в течении семестра. Для подготовки к коллоквиуму необходимо изучить материал, описанный в лекционном курсе.

Темы, которые необходимо подготовить к коллоквиуму (1 семестр):

Структурное программирование.

Синтаксис и алфавит языка Си.

Простые типы данных языка Си.

Конструкции структурного программирования.

Сложные типы данных.

Пример билета для коллоквиума представлен в приложении А.

Примерный билет

1. На псевдокоде записан алгоритм:

```

ВВОД n
ВВОД x[n]
i = 1, k = 0
пока i <= n делать
    если x[i] > 0 то k = k + 1
    i = i + 1
конец цикла
Вывод k

```

Запишите имена переменных-счетчиков _____

2. Выберите задачи, для решения которых требуется построить алгоритм, использующий ветвление:

- а) нахождение гипотенузы по двум катетам
- б) нахождение суммы элементов массива
- в) нахождение произведения двух матриц
- г) решение квадратного уравнения

3. Запишите на псевдокоде алгоритм проверки существования положительных элементов в массиве $X[n]$.

4. Чему будет равна переменная S после выполнения алгоритма:

Шаг1. $S := 0$

Шаг2. ДЛЯ i ОТ 1 ДО 3

Шаг2.1. $S := S + i$;

КОНЕЦ ЦИКЛА

Шаг3. Печать S .

5. Может ли переменная на Си иметь имя 31_p20 ?

6. Какой тип нужно использовать для описания переменной n , если ее значение используется в формуле:

$$S = \sum_{n=1}^{\infty} \frac{(-1)^n}{n!} \cdot (n+1)$$

7. Что будет выведено на экран при выполнении следующего фрагмента программы:

```

int x = 1;
int y = 2;
int z = 3;
if (x > y) { z = y * 2;
            y = x * 4; }
else { z = x * 2;
       x = y + x; }
printf(" %d %d %d", x, y, z);

```

8. Запишите фрагмент программы, решающей следующую задачу (используйте цикл for):

Вывести на экран следующие числа:

2 5 8 11 14 17 20 23

Фрагмент обязательно должен содержать описания использованных переменных.

9. Что будет выведено на экран при выполнении следующего фрагмента программы:

```

int k = 0;
while (k < 10)
    k++;
printf("%d\n", k);

```

10. Чему будет равно значение переменной f после выполнения следующего фрагмента программы

```

...
int k = 11;
int f = 0;
switch(k){
case 1: f = 1; break;
case 2: f = 2; break;
case 3: f = 3; break;
case 4: f = 4; break;
case 5: f = 5; break;
default: f = 10;
}

```

...

11. Запишите фрагмент программы, который выполняет следующие действия:

1. Опишите *z* как указатель на *float*.
2. Выделите память под *z*
3. Запишите по адресу *z* значение 3.11

12. В программе переменные *j* и *i* описаны следующим образом: *char j; int i*

Запишите функцию *printf*, выводящую на экран значения этих переменных в строке:

“Значение *i* = <...> Значение *j* = <...>”

13. В программе переменная *j* описана следующим образом: *int j;*

Запишите функцию *scanf*, считывающую значение переменной *j* с клавиатуры

14. Что будет выведено на экран при выполнении следующей программы:

```
int main(int argc, char *argv[])
{
    system("chcp 1251");
    int x[10] = {2,7,6,1,9,5,8,3,4,0};
    int k = x[0];
    for (int i=0;i<10;i++)
        if (x[i]>k) k = x[i];
    printf("%3d",k);
    printf("\n");
    system("PAUSE");
    return EXIT_SUCCESS;
}
```

15. Назовите алгоритм сортировки, который фиксирует элементы массива, начиная с первого, и ищет на зафиксированное место минимальный элемент, среди элементов с номерами от 1 до *n*. Далее фиксируется второй элемент и ищется минимальный элемент среди оставшихся, найденный элемент выставляется на второе место и т.д..

16. Элементы матрицы задаются следующим фрагментом программы:

```
int x[3][3];
int n = 3;
for (int i=0;i<n;i++)
    for(int j=0;j<n;j++)
        if(i==j) x[i][j] = i;
        else x[i][j] = -1;
```

Запишите получившуюся матрицу.

17. Опишите структуру, хранящую данные следующих типов:

I – вещественное число

M – символ

S – указатель на целое число

18. Чему равен размер переменной *p*, если в программе сделаны следующие описания:

```
union Example {
    int k;
    char z[5];
} p;
```

19. Результат какого типа возвращает функция *int f10 (float x, int y) {...}***20. Что будет выведено на экран при выполнении следующей программы:**

```
#include <cstdlib>
#include <iostream>

using namespace std;
int Function (int x, int y, int *k){
    x+=13;
    y+=10;
    (*k)+=12;
    return y;
}

int main(int argc, char *argv[])
{
    int x = 1;
    int y = 1;
    int k = 1;
    x = Function(x,y,&k);
    printf("x = %d y = %d k = %d\n",x,y,k);
    system("PAUSE");
    return EXIT_SUCCESS;
}
```