

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

«__» _____ 2015 г.

Информатика (Информатика и программирование)

методические указания к самостоятельной работе

для студентов направления

080500.62 – «Бизнес-информатика» и

231000.62 – «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. Выполнение командных заданий _____	6
8. Подготовка к коллоквиуму_____	6
Приложение А. Примерный билет_____	8
Приложение Б. Примерный билет_____	9

ВВЕДЕНИЕ

Дисциплина «Информатика» относится к обязательным дисциплинам вариативной части профессионального цикла подготовки студентов направления «Бизнес-информатика» (БИ).

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 дисциплинам «Информатика» и «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 (<i>примеры</i>)
Подготовка к контрольным работам	40	Контрольная работа
Подготовка к лабораторным работам	32	Отчет, защита лабораторной работы
Выполнение домашних заданий	36	Отчет, защита домашних заданий
Выполнение командных заданий	30	Защита командного задания
Подготовка к коллоквиуму	20	Коллоквиум
Подготовка к тестовым опросам	22	Тест
Курсовое проектирование (только БИ)	54	Пояснительная записка, защита курсового проекта
Всего часов самостоятельной работы	234 (БИ) 180 (ПИ)	

3. ТЕМЫ КОНТРОЛЬНЫХ РАБОТ

Представление данных в ЭВМ.
 Способы представления алгоритмов.
 Синтаксис языка Си.
 Циклы, проверка условий в языке Си.
 Работа с массивами в языке Си.
 Работа с матрицами в языке Си.
 Работа со строками в языке Си.
 Динамические списки.

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.
 Сортировка массивов.

Динамические матрицы.
Строки.
Обработка матриц. Функции.
Текстовые файлы.
Двоичные файлы.
Односвязные динамические списки.
Управление консолью.
Рисование графика функции.
Создание простейшего диалога.

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/Informatika_Inform_i_programmirovanie_080500_62_231000_62_file_609_5748.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.
Функции.
Файлы.
Двусвязные списки.

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. ВЫПОЛНЕНИЕ КОМАНДНЫХ ЗАДАНИЙ

Во время изучения дисциплины студентам предстоит выполнить 2 командных задания – разработка комплекса численных методов и разработка справочной системы. Для выполнения командного задания будут сформированы группы студентов из 5-6 человек. Каждой группе будет предложено индивидуальное задание. В ходе работы команда должна распределить роли участников и определить индивидуальные задачи. В результате работы команда должна представить рабочий программный продукт и техническую документацию. Работа в команде развивает навыки совместной работы, умение решать поставленные задачи во взаимодействии с коллективом, что согласуется с требованиями, предъявляемыми к освоению дисциплины.

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.

Темы, которые необходимо подготовить к коллоквиуму (2 семестр):

Функции.

Файлы.

Динамические списки.

Программирование в *WinAPI*.

Пример билета для коллоквиума представлен в приложении Б.

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;
}
```

Примерный билет

- Выберите верные утверждения:
 - Второй аргумент функции *fopen()* «w», следовательно, файл открыт для чтения как текстовый;
 - Второй аргумент функции *fopen()* «w», следовательно, файл открыт для чтения как двоичный;
 - Второй аргумент функции *fopen()* «w», следовательно, файл открыт для записи как текстовый;
 - Второй аргумент функции *fopen()* «w», следовательно, файл открыт для записи как двоичный;
- Какая функция используется для перемещения указателя чтения-записи файла? Опишите ее аргументы.
- Запишите фрагмент программы, который записывает в текстовый файл числа 2 4 6 8 10.

- В текстовом файле *data.txt* записана следующая информация:

Различия в химическом составе горных пород Земли и ее спутника «подтверждают гипотезу, о том, Луна сформировалась в результате сильнейшего столкновения прото-Земли с небесным телом», заявили ученые. Компьютерные модели, воссоздающие удар гигантской силы, показывают, что Луна состоит на 40 % из обломков Тейи, и, следовательно, имеет отличное от земных соотношение изотопов.

Чему будет равно значение переменной *C* после выполнения следующего фрагмента программы:

...

```
system("chcp 1251");
FILE *f = fopen("data.txt", "r");
char C[100];
for(int i=0; i<7; i++)
    fscanf(f, "%s", C);
printf("%s\n", C);
```

- В двоичном файле “*Examp.dat*” хранится матрица целых чисел 5Х6.

Выполните на языке Си следующие действия:

Опишите указатель на файл;

Откройте двоичный файл “*Examp.dat*” для чтения. Не считывая всю информацию, прочитайте и выведите на экран элементы первого столбца.

- В двоичном файле записана матрица

```
1 2 3 4 5
5 6 7 8 9
9 8 7 6 5
4 3 2 1 0
2 3 5 1 5
```

Что будет выведено на экран при выполнении следующего фрагмента программы:

```
int z;
int k = 4*sizeof(int);
for(int i=0; i<4; i++) {
    fseek(f, k, SEEK_SET);
    fread(&z, sizeof(int), 1, f);
```

```
k+=4*sizeof(int);
printf("%4d",z);}
```

7. Опишите структуру для создания связанного списка, хранящего информацию о имеющихся на складе товарах (название товара, цена, количество).
8. В стек добавляются элементы в следующем порядке: "1", "2", "3", "4", "5", далее выполняется следующая последовательность действий: удалить два элемента, добавить "2", удалить 2 элемента. Если стек пуст, то добавить в стек "3", в противном случае удалить один элемент. Запишите вид полученного после этих операций стека.
9. Выполните действия, описанные в предыдущем задании для очереди.

10. Функция WinMain описана в программе следующим образом:

```
int WINAPI WinMain (HINSTANCE hThisInstance, HINSTANCE hPrevInstance, LPSTR lpszArgument,
                    int nFunsterStil).
```

Что передается приложению в параметре *hThisInstance*?

11. Когда приложение получает сообщение **WM_PAINT**?

12. Опишите (изобразите) структуру меню, если в ресурсном файле выполнены следующие описания:

```
MenuEx MENU {
POPUP "Создание массива" {
MENUITEM "Читать из файла", 101
MENUITEM SEPARATOR
MENUITEM "Задать случайным образом", 102 }
POPUP "Печать", GRAYED {
MENUITEM "В файл", 103
MENUITEM "На экран", 104
}
MENUITEM "Выход", 105 }
```

13. Выберите истинные высказывания:

- Модальные диалоговые окна не дают возможности работать с другими окнами, созданными родительским окном
- Модальные окна не разрешают переключаться на работу с другими приложениями.
- У модального диалогового окна нет собственной оконной функции.
- Немодальное диалоговое окно создается с помощью функции DialogBox().

14. Перечислите элементы диалогового окна, если в ресурсном файле выполнены следующие описания:

```
Dialog5 DIALOG 30,34,152,67
STYLE DS_MODALFRAME | WS_POPUP | WS_CAPTION
CAPTION "Пример диалога"
{
    CTEXT "Значение: ",
    -1, 20, 9, 117, 26, WS_CHILD | WS_VISIBLE CONTROL "", IDNAME, "EDIT",
    ES_LEFT | WS_CHILD | WS_VISIBLE | WS_BORDER | WS_GROUP, 7, 20, 143, 12
    CONTROL "Выход", IDENTER, "BUTTON", WS_CHILD | WS_VISIBLE,
    25, 45, 50, 15
    PUSHBUTTON "Найти", IDCANCEL, 95, 45, 50, 15, WS_CHILD | WS_VISIBLE |
```

