Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Новосибирский государственный технический университет»
Кафедра автоматизированных систем управления
Кафедра автоматики
Кафедра вычислительной техники

Паспорт контрольной работы

по дисциплине «Информатика», 1 семестр

1. Методика оценки

Выполнение контрольной работы является формой текущей аттестации (контроля) по дисциплине, предусмотренной учебным планом.

Контрольная работа направлена на закрепление и проверку уровня владения учебным материалом по теоретическим темам и темам практических занятий, а также формирование навыков самостоятельного анализа процессов и явлений. Контрольная работа проводится по темам:

Выполнение тестовых заданий по темам:

1. анализ программ: "оживление" фрагментов, описание данных, логический и временной анализ;
2. функции: оформление фрагментов в виде функций, вызовы функций;
3. разработка алгоритмов решения "арифметических задач" по вариантам в соответствии с ЖЦ программы в технологии структурного программирования (образная модель, паттерны, выстраивание кода "от общего к частному).

Набор фрагментов для тем 1,2 (в количестве 15), вариант задания для темы 3 генерируются рандомно. Изменение варианта задания возможно только по согласованию с преподавателем.

Количество заданий достаточно для обеспечения каждого обучающегося заданием контрольной работы.

Структура контрольной работы:

1. Титульный лист
2. Основная часть (задание контрольной работы).
3. Заключение (выводы и рекомендации).
4. Список литературы и источников.

Основная часть – это ответ задания контрольной работы. Он должен быть самостоятельным, развернутым и аргументированным. При необходимости основная часть может быть разбита на более мелкие вопросы. Она должна содержать обязательные ссылки на изученную литературу, нормативные акты и интернет-источники, оформленные постранично в соответствии с библиографическими требованиями.

Заключение: изложение общего вывода по изученной проблеме и предлагаемых рекомендаций.

Список литературы оформляется в соответствии с библиографическими требованиями в алфавитном порядке и включает от 3 до 5 источников (книг, статей разных авторов, интернет-источников, документов), которые были изучены при выполнении контрольной работы.

Рекомендуется излагать мысли по существу, кратко и логично.

Требования к оформлению:

Объем контрольной работы до 10 страниц машинописного текста формата А4. Шрифт Times New Roman, 12. Нумерация страниц сквозная, в нижней части листа по

центру арабскими цифрами. Контрольная работа должна быть отредактирована, не содержать орфографических, синтаксических и стилевых ошибок.

Контрольная работа предоставляется для проверки в электронном виде в срок, установленный преподавателем. При положительном результате оценивания контрольной работы студент её распечатывает, передает на кафедру и защищает до сессии в назначенное преподавателем время. По результатам защиты студенту выставляется оценка в соответствии с критериями, приведенными в п. 2 настоящего Паспорта.

1. **Критерии оценки результатов обучения, соотнесенных с уровнями освоения индикаторов достижения компетенций**

Общие правила выставления оценок текущей аттестации определяются балльно-рейтинговой системой, установленной локальным актом НГТУ.

Контрольная работа выполнена **на продвинутом** уровне, если структура, содержание и оформление работы соответствует требованиям. Все части контрольной работы согласованы, текст логично выстроен и является авторским. Присутствуют ссылки на нормативные документы и актуальную литературу. Работа представлена для проверки в установленные сроки. Анализ каждого из разделов контрольной работы свидетельствует о том, что совокупность результатов ее выполнения и соотнесенных с ними индикаторов достижения компетенций не содержит пробелов. Закрепленные за контрольной работой компетенции сформированы на продвинутом уровне. Оценка составляет *от 25 до 30 баллов*.

Контрольная работа выполнена **на базовом** уровне, если структура, содержание и оформление работы соответствует требованиям, но работа содержит единичные не принципиальные ошибки, исправленные после замечаний преподавателя. Все части контрольной работы согласованы, текст логично выстроен и является авторским. Присутствуют ссылки на нормативные документы и актуальную литературу. Работа представлена для проверки в установленные сроки. Анализ каждого из разделов контрольной работы свидетельствует о том, что совокупность результатов ее выполнения и соотнесенных с ними индикаторов достижения компетенций содержит несущественные пробелы. Закрепленные за контрольной работой компетенции сформированы на базовом уровне. Оценка составляет *от 16 до24 баллов*.

Контрольная работа выполнена **на пороговом** уровне, если структура, содержание и оформление работы соответствует требованиям, но работа содержит ошибки, неоднократно исправляемые после замечаний преподавателя. Части контрольной работы в целом согласованы. Присутствуют ссылки на нормативные документы и актуальную литературу. Работа представлена для проверки в установленные сроки. Анализ каждого из разделов контрольной работы свидетельствует о том, что совокупность результатов ее выполнения и соотнесенных с ними индикаторов достижения компетенций содержит пробелы. Закрепленные за контрольной работой компетенции сформированы на пороговом уровне. Оценка составляет *от 12 до15 баллов*.

Контрольная работа считается **не выполненной,** если структура, содержание и оформление работы не соответствует требованиям, работа содержит существенные ошибки, не исправленные после замечаний преподавателя. Части контрольной работы не согласованы. Отсутствуют ссылки на нормативные документы и актуальную литературу. Работа не представлена для проверки в установленные сроки. Анализ каждого из разделов контрольной работы свидетельствует о том, что совокупность результатов ее выполнения и соотнесенных с ними индикаторов достижения компетенций не содержит множество существенных пробелов. Закрепленные за контрольной работой компетенции не сформированы. Оценка составляет *от 0 до 11 баллов*.

2. Шкала оценки

Контрольная работа как форма текущей аттестации (контроля) по дисциплине считается успешно выполненной, если сумма полученных баллов по всем ее заданиям составляет от 12 до 24 баллов включительно.

В общей оценке по дисциплине баллы за выполнение контрольной работы учитываются в соответствии с правилами балльно-рейтинговой системы и таблицей соответствия баллов, традиционной оценки и буквенной оценки ECTS, установленными в НГТУ.

3. Примерный перечень заданий (вариантов) контрольной работы

3.1. Тема 1. Анализ программ: "оживление" фрагментов, описание данных, логический и временной анализ;

```
//-----------------------------------------------------------13
for (n=a, s=0; n!=0; n=n/10)
    { k=n%10; s=s+k;}
//-----------------------------------------------------------14
for (n=a, s=0; n!=0; n=n/10)
    { k=n%10; if (k>s) s=k;}
//-----------------------------------------------------------15
for (n=a, s=0; n!=0; n=n/10)
    { k=n%10; s=s*10+k;}
//-----------------------------------------------------------16
for (i=0, n=a; n!=0; i++, n=n/10);
for (A[i--]=-1, n=a; n!=0; i--, n=n/10)
    A[i]=n % 10;
//-----------------------------------------------------------17
for (j=0,a=10; a<30000; a++){
                for (n=a, s=0; n!=0; n=n/10)
                { k=n%10; s=s+k;}
   if (a==s*s*s) A[j++]=a;
}
//-----------------------------------------------------------18
for (j=0,a=10; a<v; a++){
                for (n=a, s=0; n!=0; n=n/10)
        { k=n%10; s=s*10+k;}
   if (a==s) A[j++]=a;
   }
//-----------------------------------------------------------19
for (i=0; i<n; i++){
   if (A[i]<0){
                for (j=i; j<n-1;j++) A[j]=A[j+1];
                n--;
                i--;
                }}
//-----------------------------------------------------------20
for (i=1,k=0; i<n; i++)
   if (A[i]>A[k]) k=i;
for (j=k; j<n-1;j++) A[j]=A[j+1];
   n--;
//-----------------------------------------------------------21
for (i=0,a=2; a<v; a++){
                for (n=2; n<a; n++)
        { if (a%n==0) break; }
                if (n==a) A[i++]=a;
                }
```

```
A[i]=0;
//-----------------------------------------------------------22
for (i=0,a=2; a<v; a++){
                    for (s=0,n=2; n<a; n++)
                    if (a%n==0) { s=1; break; }
                    if (s==0) A[i++]=a;
                    }
A[i]=0;
//-----------------------------------------------------------23
for (j=0,i=0; i<n ; i++){
                    for (m=2; m<A[i]; m++)
        { if (A[i]%m==0) break; }
                    if (m==A[i]) B[j++]=A[i];
                    }
B[j]=0;
//-----------------------------------------------------------24
for (j=0,i=0; i<n ; i++){
                    for (s=0,m=2; m<A[i]; m++)
        if (A[i]%m==0) { s=1; break; }
                    if (s==0) B[j++]=A[i];
                    }
B[j]=0;
//-----------------------------------------------------------25
for (i=0; i<n ; i++){
                    for (m=2; m<A[i]; m++)
        { if (A[i]%m==0) break; }
                    if (m==A[i]) {
                    for (j=i; j<n-1;j++) A[j]=A[j+1];
                    n--;
                    i--;
                    }}
//-----------------------------------------------------------26
for (j=0,i=0; i<n ; i++){
                    for (s=0,m=2; m<A[i]; m++)
        if (A[i]%m==0) { s=1; break; }
                    if (s==0) {
                    for (j=i; j<n-1;j++) A[j]=A[j+1];
                    n--;
                    i--;
                    }}
//-----------------------------------------------------------27
for (i=0; i<n-1 && val !=1; i++){
                    for (m=2; val % m !=0; m++);
                    val /= m; A[i] = m;
                    }
A[i] = 0;
//-----------------------------------------------------------28
v=A[0]+1;
do                  {
   v--;
   for (i=0,s=0;i<n;i++)
                    if (A[i]%v!=0) { s=1; break; }
   } while(s==1);
//-----------------------------------------------------------29
for (i=0; i<n; i++)
   if (A[i]%v!=0) { v--; i=-1; }
//-----------------------------------------------------------30
for (i=0,a=2; a<v && i<m-1 ; a++){
                    for (s=0,j=0; j<i; j++)
```

```
                    if (a%A[j]==0) { s=1; break; }
                    if (s==0) A[i++]=a;
                    }
A[i]=0;
//-----------------------------------------------------------31
for (i=0; i<n-1; i++)
for (j=i+1; j<n; j++)
        if (c[i]==c[j]) return i;
return -1;
//-----------------------------------------------------------32
for (s=0,i=0; i<n; i++){
   for (k=0,j=0; j<n; j++)
         if (c[i]==c[j]) k++;
   if (k>s) s=k,b=i;
   }
//-----------------------------------------------------------33
for (s=0,i=0; i<n-1; i++)
  if (A[i]==A[i+1]){
   for (k=2; i+k<n && A[i]==A[i+k]; k++);
   if (k>s) s=k,b=i;
   }
//-----------------------------------------------------------34
for (k=0, m=1; m <= n; k++, m = m * 2);
return k-1; }
//-----------------------------------------------------------35
for (i=0,j=n-1; i < j; i++,j--)
    { k = c[i]; c[i] = c[j]; c[j] = k; }
//-----------------------------------------------------------36
for (i=0; i<n; i++){
   for (j=k1=k2=0; j<n; j++)
        if (c[i] != c[j])
                { if (c[i] < c[j]) k1++; else k2++; }
   if (k1 == k2) return i;
   }
return -1;
//-----------------------------------------------------------37
for (s=0, i=0; i<n-1; i++){
                    for (j=i+1, m=0; j<n; j++)
                    if (c[i]==c[j]) m++;
                    if (m > s) s=m, b=i;
                    }
//-----------------------------------------------------------38
for (i=k=m=0; i<n-1; i++)
   if (c[i]==c[i+1]) k++;
                    else        {
                                if (k > m) m=k, b=i-k-1;
                                k=0;
                                }
//-----------------------------------------------------------39
for (s=0,i=0; i<n; i++){
   if (A[i]<0) continue;
   if (A[i]==0) break;
   s=s+A[i];
   }
//-----------------------------------------------------------40
for (s=0,i=0; i<n && A[i]>0; i++)
   s=s+A[i];
//-----------------------------------------------------------41
for (k=0,s=0,i=0; i<n && k==0; i++){
```

```
    if (A[i]<0) k=1;
    s=s+A[i];
    }
//-----------------------------------------------------------42
for (s1=0,s2=0,i=0,j=n-1; i<=j;){
    if (s1<s2) s1+=A[i],i++;
    else s2+=A[j],j--;
    }
return i;
//-----------------------------------------------------------43
for (s=0,i=0; i<n; i++){
    if (i%2==0) s=s+A[i];
    else s=s-A[i];
    }
//-----------------------------------------------------------44
for(j=0;n!=0;j++){
    for (k=0,i=1; i<n; i++)
                    if (A[i]<A[k]) k=i;
    B[j]=A[k];
    for (;k<n-1;k++) A[k]=A[k+1];
    n--;
    }
//-----------------------------------------------------------45
for(j=0,max=A[0];j<n;j++)
    if (A[j]>max) max=A[j];
for(j=0;j<n;j++){
    for (k=0,i=1; i<n; i++)
                    if (A[i]<A[k]) k=i;
    B[j]=A[k];
    A[k]=max+1;
    }
//-----------------------------------------------------------46
while(n!=0){
    for (k=0,i=1; i<n; i++)
                    if (A[i]<A[k]) k=i;
    c=A[k]; A[k]=A[n-1]; A[n-1]=c;
    n--;
    }
//-----------------------------------------------------------47
for (j=0,a=10; a<v; a++){
                    for (s=0,n=a, s=0; n!=0; n=n/10){
      k=n%10;
      if (k!=0 && a%k!=0) { s=1; break; }
      }
    if (s==0) A[j++]=a;
    }
//-----------------------------------------------------------48
for (i=0; i<n-1; i++)
   if (A[i]==A[i+1]){
    for (j=i; j<n-2;j++) A[j]=A[j+2];
    n=n-2;
    i--;
    }
//-----------------------------------------------------------49
for (i=0,k=-1; i<10; i++){
    if (A[i]<0) continue;
    if (k==-1) k=i;
    else
                    if (A[i]<A[k]) k=i;
```

```
   }
//-----------------------------------------------------------50
for (i=0,s=0,k=0; i<10; i++)
    if (A[i]<0) k=1;
    else {
    if (k==1) s++;
    k=0;
    }
//-----------------------------------------------------------51
for (i=0,s=0; i<10; i++)
    if (A[i]>s) s=A[i];
//-----------------------------------------------------------52
for (i=1,k=0; i<10; i++)
    if (A[i]>A[k]) k=i;
//-----------------------------------------------------------53
for (i=0,k=-1; i<10; i++)
{ if (A[i]<0) continue;
if (k==-1) k=i;
else
    if (A[i]<A[k]) k=i;
}
//-----------------------------------------------------------54
for (i=0,k=-1; i<10; i++)
{ if (A[i]<0) continue;
if (k==-1 || A[i]<A[k]) k=i;
}
//-----------------------------------------------------------55
for (i=0,s=0; i<10; i++)
    if (A[i]>0) s++;
//-----------------------------------------------------------56
for (i=1,s=0; i<10; i++)
    if (A[i]>0 && A[i-1]<0) s++;
//-----------------------------------------------------------57
for (i=1,s=0,k=0; i<10; i++){
if (A[i-1]<A[i]) k++;
else{
if (k>s) s=k;
                    k=0;
                    }}
//-----------------------------------------------------------58
for (i=0,s=0,k=0; i<10; i++)
    if (A[i]<0) k=1;
else
    { if (k==1) s++; k=0; }
```

3.2.Тема 2. Функции: оформление фрагментов в виде функций, вызовы функций;

Задания аналогичны 3.1.

3.3.Тема 3. Разработка алгоритмов решения "арифметических задач" по вариантам в соответствии с ЖЦ программы в технологии структурного программирования (образная модель, паттерны, выстраивание кода "от общего к частному).

1. Найти в массиве и вывести значение наиболее часто встречающегося элемента.

2. Найти в массиве элемент, наиболее близкий к среднему арифметическому суммы его элементов.

3. Найти наименьшее общее кратное всех элементов массива (то есть числа, которое делится на все элементы).

4. Найти наибольший общий делитель всех элементов массива (на который они все делятся без остатка).

5. Интервал между минимальным и максимальным значениями элементов массива разбить пополам и относительно этого значения разделить массив на две части – меньшие слева, большие справа (части не сортировать).

6. Задан массив, определить значение k, делящее массив на две части так, чтобы разность сумм (по модулю) справа и слева была минимальна («центр тяжести» массива).

7. Найти в массиве наибольшее число подряд идущих одинаковых элементов (например {1,5,3,6,6,6,6,6,3,4,4,5,5,5} = 5).

8. Найти в массиве все пары одинаковых, рядом стоящих элементов и удалить.

9. Удалить из массива все отрицательные элементы.

10. Найти в массиве все пары взаимно простых элементов (не обязательно рядом стоящих) и вывести.

11. Найти в массиве все последовательности возрастающих элементов и копировать в выходной массив.

12. Удалить из массива все не простые числа (оставить простые)

13. Найти в массиве все пары элементов (не обязательно рядом стоящих), у которых последняя цифра первого элемента равна первой цифре второго и вывести.

14. В массиве, содержащем нечетное количество элементов, найти медиану значение, для которого количество больших элементов равно количеству меньших. Учесть тот факт, что при наличии равных элементов медиана не может быть определена.

15. Определить количество неповторяющихся значений элементов массива (например 1,7,2,8,1,1,1,2,5,7,7 -> 2).

16. Найти в массиве минимальный и максимальный элементы. Вывести последовательность значений из этого диапазона, не встречающихся в данном массиве.

17. Выделить в массиве возрастающую последовательность, начиная с первого элемента и перенести ее в другой массив (например, **3** ,2,1, **4** , **6** ,3, **8** ,1,1, **12** -> 3,4,6,8,12).