

Фонд оценочных средств (контрольно-измерительные материалы)

Министерство просвещения Российской Федерации

федеральное государственное бюджетное образовательное учреждение высшего образования

«Красноярский государственный педагогический университет
им. В.П. Астафьева»

Институт математики, физики и информатики

(наименование института/факультета)

Кафедра-разработчик Информатики и информационных технологий в
образовании

(наименование кафедры)

УТВЕРЖДЕНО

на заседании кафедры

Протокол № 9

от «08» мая 2024 г.

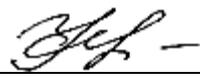
ОДОБРЕНО

на заседании научно-методического совета

направления подготовки

Протокол № 7 от «15» мая 2024 г.


Пак Н.И.


Аёшина Е.А.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

для проведения текущего контроля и промежуточной аттестации обучающихся

«Программирование и робототехника»

(наименование дисциплины/модуля/вида практики)

Направление подготовки:

44.04.01 Педагогическое образование

направленность (профиль) образовательной программы

Технологии цифровизации образовательной деятельности

(с применением сетевой формы)

по заочной форме обучения (общая трудоемкость дисциплины б з.е.)

Составители:

канд. пед. наук, доцент кафедры ИИТвО Яшина И.А..

3.3.1. Оценочное средство 1 «Экзаменационный тест»

- 1) Какой язык из перечисленных наиболее эффективен для программирования искусственного интеллекта?
 - a) LISP
 - b) FORTRAN
 - c) ALGOL
- 2) Какие языки относятся к языкам объектно-ориентированного программирования?
 - a) FORTRAN.
 - b) Java.
 - c) C++.
 - d) LISP.
 - e) Python.
 - f) Pascal.
- 3) Какой из циклов оптимальней использовать для повторения оператора(ов), если известно количество шагов цикла?
 - a) Цикл "while".
 - b) Цикл "repeat Until".
 - c) Цикл "for".
- 4) Какие языки относятся к алгоритмическим языкам?
 - a) LISP.
 - b) PERL.
 - c) Pascal.
- 5) Какой из циклов нужно использовать, если известно, что тело цикла должен выполняться хотя бы один раз?
 - a) Цикл "while".
 - b) Цикл "repeat Until".
 - c) Цикл "for".
- 6) Для того чтобы вывести символ новой строки, надо:
 - a) Закончить оператор точкой с запятой.
 - b) Воспользоваться специальным макросом endl
 - c) При выводе строки символов перевод строки добавляется автоматически.
- 7) Компилятор языка Си++:
 - a) Переводит текст программы в машинные инструкции.
 - b) Выполняет программу.
 - c) Форматирует текст программы так, чтобы его было удобно читать.
- 8) Комментарий в программе на Си++
 - a) Содержит указания компилятору по настройке программы.

- b) Содержит пояснения к тексту и не оказывает влияния на выполнение программы.
 - c) Должен содержать допустимые аргументы программы.
- 9) **При выходе из функции main**
- a) Программа повторяется с теми же аргументами.
 - b) Программа заканчивается.
 - c) Выполняется функция finish, определенная программистом.
- 10) **Объявление переменной**
- a) Необходимо сделать до того, как использовать эту переменную.
 - b) Можно сделать в любой момент до завершения работы программы.
 - c) Желательно сделать для лучшего понимания программы, но можно и опустить.
- 11) **Выберите правильное объявление константы pi:**
- a) `const float pi = 3.14;`
 - b) `float pi = (const) 3.14;`
 - c) `const float pi; pi = 3.14;`
- 12) **Отметьте правильное определение константы:**
- a) `const long BITS = 32;`
 - b) `const bit ZERO = 0x0;`
 - c) `const float 0 Ora = 5.9787;`
- 13) **Укажите пункты, в которых происходит объявление констант и переменных**
- a) `float dD;`
 - b) `float sd2 = 3.2;`
 - c) `bType = 3;`
 - d) `int k; k = 89;`
- 14) **Укажите, в каких выражениях используются ключевые слова?**
- a) `sdf = 2; int r = 24;`
 - b) `TStringList *S = new TStringList;`
 - c) `x = 3; x = x + 4;`
 - d) `void function()`
- 15) **Если после выражения стоит точка с запятой, то**
- a) Это оператор-выражение, действие которого заключается в вычислении выражения.
 - b) Выражение вычисляется, а его значение запоминается в специальной переменной, которую можно использовать в следующем операторе.
 - c) Выражение вычисляется только если первой стоит операция присваивания.

16) Чему равен результат вычисления выражения $x + 3*b + x$ при $x = 12$ и $b = 8$?

- a) 132
- b) 48
- c) 300

17) Каково будет значение переменной k после выполнения оператора $k = ++k$; если до его выполнения k равнялось 6?

- a) 6
- b) 7
- c) 8

18) Если $\text{int } i=3$, какой будет результат вычисления:

```
if (i == 4) cout << "aaa";  
else if (i == 3) cout << "bbb";  
else if (i != 3) cout << "ccc";
```

- a) aaa
- b) bbb
- c) ccc
- d) aaaccc
- e) bbbccc
- f) ошибка компиляции

19) Что выведет на экран следующая программа ?

```
#include <iostream. h>  
int main () {  
    int i;  
    for(i = 0; i < 9; i++)  
        cout << i+1;  
    return 1;  
}
```

- a) Цифры от 0 до 8.
- b) Цифры от 1 до 9.
- c) Программа не будет построена из-за ошибок.

20) Укажите, какой будет результат вычисления k ?

```
int func()  
{  
    int k = 10;  
    for (int i = 0; i <= k; i++)  
    {  
        return;  
        k = i;  
    }  
}
```

```
    }  
}
```

- a) 0
- b) 10
- c) 1
- d) бесконечный цикл
- e) 100

21) Если `int n=45`, какой будет результат?

```
switch(n) {  
case 23: cout << "aaa";  
case 45: cout << "bbb";  
default: cout << "vvv";  
break; }
```

- a) ошибка компилятора
- b) aaa
- c) bbb
- d) vvv
- e) bbbvvv
- f) aaavvv
- g) Неопределенное поведение.

22) При выполнении фрагмента кода

```
int x = 3, y = 2, z = 1;  
if(x >= y)  
if(y <= z)  
cout << "Вариант 1";  
else  
if( x >= z)  
cout << "Вариант 2";  
else  
cout << "Вариант 3";  
будет напечатано:
```

- a) Вариант 1
- b) Вариант 2
- c) Вариант 3
- d) Ничего не будет напечатано
- e) Программа не откомпилируется

23) Укажите, каким будет значение `k` в результате вычислений?

```
int func()  
{
```

```
int k = 10;
for (int i = 0; i <= k; i++)
{
return;
k = i; }
}
```

- a) 0
- b) 10
- c) 1
- d) Бесконечный цикл.
- e) 100

24) Если `int n=3`, какой будет результат?

```
switch(n) {
case 2: cout << "aaa"; break;
case 3: cout << "bbb"; break;
default: cout << "vvv"; break; }
```

- a) ошибка компилятора
- b) aaa
- c) bbb
- d) vvv
- e) неопределенное поведение

25) Что произойдет при выполнении:

```
for ( int i = 0; i < 5;) {
continue; i--; func(); }
```

- a) функция `func` выполнится 5 раз
- b) функция `func` не выполнится ни разу
- c) функция `func` будет выполняться бесконечно

26) Укажите, какие операторы используются для циклического выполнения кода?

- a) `if (...) then ... else ...`
- b) `while (...) {...}`
- c) `switch (...) {...}`
- d) `for (...; ...; ...) {...}`

27) Укажите результат вычисления цикла:

```
int k = 0;
do
{
k++;
if (k == 1) continue;
```

```
else break;
++k;
} while (k < 5);
```

- a) k = 2;
- b) k = 0;
- c) k = 5;
- d) k = 4;

28) Какой из наборов перечисляемых значений записан правильно?

- a) enum { a, b, 3, 4 };
- b) enum { a, b = 3, c = 4, d = 3 };
- c) enum { a, b = 3, c, d };

29) Каким будет результат работы следующего фрагмента программы:

```
int a = 5^3;
float b = 1.5f;
b += -- a/2;
cout << b;
```

- a) 63.50
- b) 64.00
- c) 3.50
- d) 4.00
- e) 63.00

30) Найдите недопустимую запись символьной константы:

- a) 'F'
- b) "\022"
- c) 'Ю'
- d) '\ a'

31) Какой результат вычисления следующего выражения?

- a) 0xFF & 5 >> 1 + 1
- b) 1
- c) 2
- d) 256

32) Каким будет результат следующей программы:

```
int a = 5*3;
float b = 1.5f;
b += -- a/2;
cout << b;
```

- a) 8.5
- b) 9.0
- c) 8.0

- d) 9.5
 - e) 7.5
- 33) **Переменные, создаваемые в теле функции, называются локальными, потому что:**
- a) Они существуют только во время выполнения функции.
 - b) К ним можно обращаться только после их объявления.
 - c) Их имена должны начинаться с символа l.
- 34) **Отметьте допустимые имена функций:**
- a) `_This_Function_12_x_`
 - b) `ax%u7`
 - c) `ffffffAAAAAA`
 - d) `3_pi`
 - e) `calculateIt`
 - f) `thisname_is_too_long_forafunction`
- 35) **Как называется функция, которая вызывает саму себя?**
- a) конструктор
 - b) деструктор
 - c) подставляемая
 - d) рекурсивная
- 36) **В чем разница между фактическими и формальными параметрами?**
- a) Формальные параметры могут использоваться только вне тела функции, а фактические - используются как вне функции, так и внутри её.
 - b) Формальные параметры определены в теле функции, а фактические - значения, с которыми функция вызывается.
 - c) Нет различий.
- 37) **Укажите правильное объявление функции**
- a) `int MyFunc(double x, y, int i)`
 - b) `void correct(double d = 3.14, double a)`
 - c) `int sum(int j, int k, int x = 0)`
- 38) **Какое из следующих утверждений об операторе `return` является верным?**
- a) Оператор `return` должен стоять последним в теле функции.
 - b) Оператор `return` завершает выполнение функции.
 - c) В теле функции должен присутствовать только один оператор `return`.
- 39) **Имеется следующее описание:**
- Type
U = ^Zveno;
Zveno = Record
X, Y : Boolean;
Pred, Next : U

```
End;  
Var  
Logic : Boolean;  
A, B : Pointer;  
X, Y : U;
```

К ошибке компиляции "Несовместимость типов" приведет следующее присваивание:

- a) $A := X^{.Next^{.Next}}$
- b) $X := Y$
- c) $Logic := X^{.X}$
- d) $X^{.Next} := A$
- e) $X^{.} := Y^{.Next}$

40) Список объявлен следующим образом:

```
Type  
Ukaz = ^Zveno;  
Zveno = Record  
X : String;  
N : Ukaz  
End;  
Var  
First : Ukaz; {ссылка на начало списка}  
В следующем фрагменте программы:  
P := First;  
While P^{.N} <> Nil Do  
Begin  
  B := P; M := P;  
  While B <> Nil Do  
  Begin  
    If B^{.X} < M^{.X} Then  
      M := B; B := B^{.N}  
  End;  
  S := P^{.X};  
  P^{.X} := M^{.X};  
  M^{.X} := S;  
  P := P^{.N}  
End;
```

выполняется

- a) Перемещение компонента к началу списка;
- b) Сортировка компонентов списка в порядке возрастания;

- c) Сортировка компонентов списка в порядке убывания;
- d) Перестановка соседних компонентов списка;
- e) Добавление в список нескольких новых компонент.

41) **Список объявлен следующим образом:**

Type

SS = ^List;

List = Record

A : LongInt;

Next : SS

End;

В приведённом фрагменте программы (First — ссылка на первый элемент списка)

P := First;

S := 0;

While Not (P = Nil) Do

Begin

S := S + 1;

P := P^.Next

End;

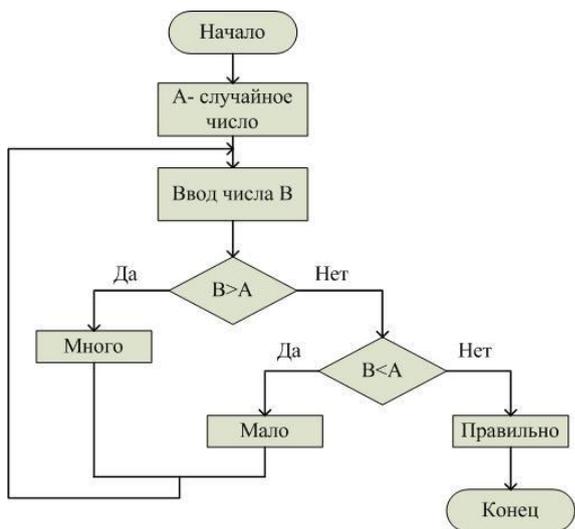
определяется

- a) Первый элемент списка;
- b) Сумма элементов списка;
- c) Сумма первого и последнего элементов списка;
- d) Количество элементов списка;
- e) Количество звеньев списка, где указатель на следующее звено не nil.

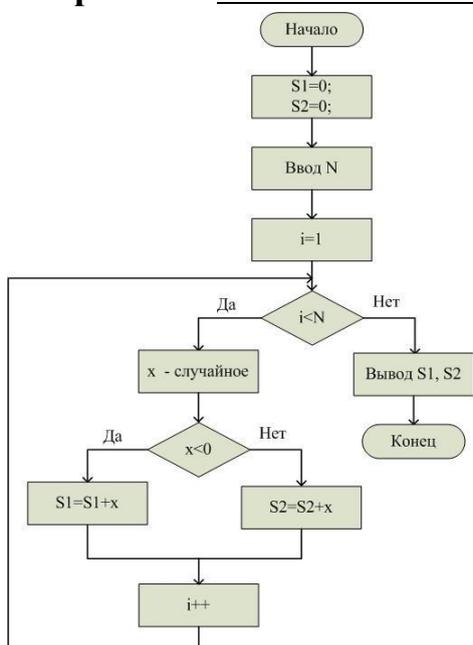
42) **Даны натуральные числа a , b , обозначающие соответственно числитель и знаменатель дроби. Сократить дробь, т.е. получить правильную несократимую дробь. Какую часть в предложенной задаче вы можете выделить как функцию? _____**

43) **Даны две дроби A/B и C/D (A, B, C, D — натуральные числа). Составить программу для деления дроби на дробь. Результат должен быть несократимой дробью. Какую часть в предложенной задаче вы можете выделить как подпрограмму? _____**

44) **Проанализировав схему, предположите, решение какой задачи отображено. _____**



45) Проанализировав схему, предположите, решение какой задачи отображено.



3.3.2. Вопросы к экзамену

- 1) Платформа Ардуино — основные принципы проектирования, сборки и программирования устройств.
- 2) Современная образовательная робототехника.
- 3) Элементная база платформы Ардуино.
- 4) Язык C/C++. Структура программы. Типы данных.
- 5) Язык C/C++. Библиотеки функций. Ввод-вывод данных. Примеры.
- 6) Язык C/C++. Организация ветвлений. Конструкция множественного выбора (switch). Пример.
- 7) Язык C/C++. Организация циклов с постусловием и предусловием. Примеры.
- 8) Язык C/C++. Цикл с параметром. Его особенности и возможности. Примеры.
- 9) Язык C/C++. Пользовательские функции. Способы передачи данных в функцию. Примеры.
- 10) Язык C/C++. Одномерные массивы. Создание массивов и обработка элементов массивов.
- 11) Передача массивов в функцию. Примеры.
- 12) Язык C/C++. Строки. Обработка строк. Связь строк, массивов и указателей. Пример.
- 13) Язык C/C++. Структуры. Описание структуры в программе. Массивы структур. Пример.
- 14) Язык C/C++. Алгоритмы поиска и сортировки
- 15) Функциональное программирование. Принципы функционального программирования. Примеры.
- 16) Язык Питон. Основные типы данных. Достоинства и недостатки. Примеры.
- 17) Язык Питон. Числовые типы данных. Операции. Свойства.
- 18) Язык Питон. Тип данных list, его возможности. Пример.
- 19) Язык Питон. Тип данных dict, его возможности. Пример.
- 20) Язык Питон. Тип данных str, его возможности. Пример.
- 21) Язык Питон. Тип данных set, его возможности. Пример.
- 22) Язык Питон и функциональное программирование. Декораторы. map. reduce. Примеры.
- 23) Сортировка в языке Питон. sort, sorted. Использование ключей сортировки.