

**Комплект оценочных материалов по дисциплине
«Программирование и основы алгоритмизации систем управления»**

Задания закрытого типа

Задания закрытого типа на выбор правильного ответа

Выберите один правильный ответ

1. Что произойдет с этим кодом на C++, если запустить его компиляцию.

```
int f (int k) {  
    k *= k;  
    return f(k);  
}  
int main () {  
    f(2);  
    return 0;  
}
```

- А) ошибка компиляции программы
- Б) ошибка через некоторое время работы программы
- В) программа скомпилируется и будет работать нормально
- Г) результат не предсказуем

Правильный ответ: Б

Компетенции (индикаторы) ОПК-6, ОПК-14

2. Укажите полное название типа `int` в языке C++

- А) `signed long int`
- Б) `unsigned long int`
- В) `signed long long int`
- Г) `unsigned long long int`
- Д) `integer`

Правильный ответ: А

Компетенции (индикаторы) ОПК-6, ОПК-14

3) Как называется процесс преобразования данных из одного типа в другой?

А) приведение типов

Б) трансформация типов

В) конвертация типов

Г) модификация типов

Правильный ответ: А

Компетенции (индикаторы) ОПК-6

4. Что такое переменная?

А) неизменяемая константа

Б) имя для значения в памяти

В) тип данных

Г) условный оператор

правильный ответ: Б

Компетенции (индикаторы) ОПК-6

5. Укажите верное объявление массива в стиле C++.

А) `int a[3] = { 5, -12, -12, 9, 10 };`

Б) `int a[] = { 5, -12, -12, 9, 10 };`

В) `int a[3] = [5, -12, -12];`

Г) `int a = [5, -12, -12];`

Правильный ответ: Б

Компетенции (индикаторы) ОПК-6, ОПК-14

6. Какую структуру данных применить для представления иерархической информации, где каждый узел связан с другими узлами без циклов?

- А) массив
- Б) стек
- В) очередь
- Г) дерево

Правильный ответ: Г

Компетенции (индикаторы) ОПК-6, ОПК-14

7. Какая проблема может возникнуть при произвольном методе динамического распределения памяти и приводит к невозможности выделения большого блока памяти, несмотря на наличие свободной памяти?

- А) утечка памяти
- Б) переполнение стека
- В) фрагментация памяти
- Г) дефрагментация

Правильный ответ: В

Компетенции (индикаторы) ОПК-6, ОПК-14

8. Как называется функция, которая не возвращает никакого значения и имеет тип void?

- А) конструктор
- Б) процедура
- В) деструктор
- Г) метод

Правильный ответ: Б

Компетенции (индикаторы) ОПК-6

Задания закрытого типа на установление соответствия

Установите правильное соответствие.

Каждому элементу левого столбца соответствует только один элемент правого столбца.

1. Установите соответствие между определениями и терминами.

| Формулировка | Термин |
|--|-------------------|
| 1) значение, по которому находится значение самой переменной | А) указатель |
| 2) переменная, в которой хранится адрес другой переменной | Б) разыменованное |
| 3) значение, полученное по адресу переменной | В) адрес |

Правильный ответ

| | | |
|---|---|---|
| 1 | 2 | 3 |
| В | А | Б |

Компетенции (индикаторы) ОПК-14

2. Как изменится адрес переменной после операции над указателем

| Операция над указателем | Результат |
|---|--|
| 1) инкремент указателя типа double | А) увеличит значение адреса на 8 байт |
| 2) прибавление к указателю типа char числа 2 | Б) увеличит значение адреса на 4 байта |
| 3) прибавление к указателю типа short результата сложения чисел 1 и 1 | В) увеличит значение адреса на 2 байта |

Правильный ответ

| | | |
|---|---|---|
| 1 | 2 | 3 |
| А | В | Б |

Компетенции (индикаторы) ОПК-14

3. Какую рекомендацию можно дать по каждому оператору цикла

| Рекомендация | Оператор |
|--|-------------|
| 1) не стоит использовать этот оператор без особой необходимости, так как он приводит к усложнению понимания кода и делает код менее читаемым | А) do while |
| 2) когда количество итераций цикла заранее неизвестно и зависит от выполнения условия | Б) while |
| 3) когда известно количество итераций или необходимо выполнить инициализацию перед циклом и итерацию после каждой итерации | В) for |
| 4) когда необходимо, чтобы блок кода выполнялся хотя бы один раз, прежде чем проверять условие | Г) goto |

Правильный ответ

| | | | |
|---|---|---|---|
| 1 | 2 | 3 | 4 |
| Г | Б | В | А |

Компетенции (индикаторы) ОПК-14

4. Установите соответствие между логическими операциями и их определениями

| Рекомендация | Оператор |
|---|--------------------|
| 1) операция, которая истинна, если оба операнда истинны | А) логическое И |
| 2) операция, которая истинна, если хотя бы один операнд истинен | Б) логическое ИЛИ |
| 3) операция, которая истинна, если только один операнд истинен | В) логическое НЕ |
| 4) операция, которая истинна, если операнд ложен | Г) исключаящее ИЛИ |

Правильный ответ

| | | | |
|---|---|---|---|
| 1 | 2 | 3 | 4 |
| А | Б | Г | В |

Компетенции (индикаторы) ОПК-14

5. Установите соответствие между определениями и терминами в объектно-ориентированном программировании.

| Формулировка | Термин |
|--|-----------------|
| 1) абстракция, описывающая методы, свойства, ещё не существующих объектов | А) класс |
| 2) конкретное представление абстракции, имеющее свои свойства и методы | Б) объект |
| 3) сгруппированные под одним именем одна или несколько переменных, возможно различных типов | В) метод класса |
| 4) функция, определённая внутри класса, которая описывает действия, выполняемые над объектами этого класса | Г) структура |

Правильный ответ

| | | | |
|---|---|---|---|
| 1 | 2 | 3 | 4 |
| А | Б | Г | В |

Компетенции (индикаторы) ОПК-6, ОПК-14

6. Установите соответствие между определениями и терминами в объектно-ориентированном программировании.

| Формулировка | Термин |
|--|----------------|
| 1) особый тип метода класса, который автоматически вызывается при создании объекта этого же класса | А) деструктор |
| 2) специальный тип метода класса, который выполняется при удалении объекта класса | Б) конструктор |
| 3) специальный метод в классе, который используется для установки значения приватного поля | В) геттер |
| 4) специальный метод в классе, который используется для получения значения приватного поля | Г) сеттер |

Правильный ответ

| | | | |
|---|---|---|---|
| 1 | 2 | 3 | 4 |
| Б | А | Г | В |

Компетенции (индикаторы) ОПК-6, ОПК-14

7. Установите соответствие между определениями и терминами в объектно-ориентированном программировании.

| Формулировка | Термин |
|--|-----------------|
| 1) свойство, позволяющее объединить в классе и данные, и методы, работающие с ними и скрыть детали реализации от пользователя | А) полиморфизм |
| 2) свойство, позволяющее создать новый класс-потомок на основе уже существующего, при этом все характеристики класса родителя присваиваются классу-потомку | Б) наследование |
| 3) свойство классов, позволяющее использовать объекты классов с одинаковым интерфейсом без информации о типе и внутренней структуре объекта | В) инкапсуляция |

Правильный ответ

| | | |
|---|---|---|
| 1 | 2 | 3 |
| В | Б | А |

Компетенции (индикаторы) ОПК-14

8. Установите соответствие между определениями и терминами в программировании структур данных.

| Формулировка | Термин |
|---|-----------------------|
| 1) Структура данных, состоящая из набора узлов, где каждый узел содержит данные и ссылку на следующий узел | А) очередь |
| 2) Структура данных, в которой элементы добавляются и удаляются с одного конца, следуя принципу FIFO (первый пришел – первый ушел) | Б) стек |
| 3) Структура данных, в которой элементы добавляются в конец, а удаляются из начала, следуя принципу LIFO (последний пришел - первый ушел) | В) массив |
| 4) Непрерывный блок памяти, состоящий из фиксированного количества элементов одного типа, доступ к которым осуществляется по индексу | Г) односвязный список |

Правильный ответ

| | | | |
|---|---|---|---|
| 1 | 2 | 3 | 4 |
| Г | А | Б | В |

Компетенции (индикаторы) ОПК-14

Задания закрытого типа на установление правильной последовательности

Установите правильную последовательность.

Запишите правильную последовательность букв слева направо.

1. Установите правильную последовательность при создании функции на языке C++:

А) определение типа возвращаемого значения

Б) указание имени функции

В) указание параметров функции

Г) реализация тела функции

Правильный ответ: А, Б, В, Г

Компетенции (индикаторы) ОПК-14

2. Расположите типы данных в порядке увеличения размера занимаемой памяти в байтах для архитектуры x86:

А) short

Б) char

В) double

Г) int

Правильный ответ: Б, А, Г, В

Компетенции (индикаторы) ОПК-14

3. Укажите правильную последовательность отдельных строчек кода на языке

C++:

А) `int main ()`

Б) `#include <iostream>`

В) `cout << 12;`

Г) `using namespace std;`

Правильный ответ: Б, Г, А, В

Компетенции (индикаторы) ОПК-14

4. Расположите следующие языки программирования в хронологическом порядке их создания, начиная с самого раннего:

А) C++

Б) C

В) C#

Г) Go

Правильный ответ: Б, А, В, Г

Компетенции (индикаторы) ОПК-14

5. Установите правильную последовательность действий при работе с динамической памятью в C++:

А) выделение памяти (`new`)

Б) использование выделенной памяти

В) освобождение памяти (`delete`)

Г) объявление указателя

Правильный ответ: Г, А, Б, В

Компетенции (индикаторы) ОПК-14

6. Укажите последовательность структур данных по возрастанию максимально возможного количества связей каждого элемента:

- А) дерево
- Б) очередь
- В) двусвязный список
- Г) сеть

Правильный ответ: Б, В, А, Г

Компетенции (индикаторы) ОПК-14

7. Какая последовательность действий соответствует правильному применению полиморфизма при разработке программы:

- А) определение базового класса с виртуальными методами
- Б) создание производного класса, переопределяющего виртуальные методы
- В) использование объектов производных классов через ссылки на базовый класс

Правильный ответ: А, Б, В

Компетенции (индикаторы) ОПК-14

8. Укажите правильную последовательность отдельных строчек кода на языке С++, если программа сначала объявляет функцию `function`, потом в главной функции объявляет целочисленную переменную, запрашивает значение переменной у пользователя, вычисляет значение квадрата этой переменной, а после главной функции, определяет функцию `function`:

- А) `int function(int a) { return a * a; }`
- Б) `int a;`
- В) `function(a);`
- Г) `int function(int);`

Правильный ответ: Г, Б, В, А

Компетенции (индикаторы) ОПК-14

Задания открытого типа

Задания открытого типа на дополнение

Напишите пропущенное слово (словосочетание)

1. Процесс преобразования исходного кода программы на языке C++ в исполняемый машинный код, который может быть запущен на компьютере – _____.

Правильный ответ: компиляция

Компетенции (индикаторы) ОПК-14

2. Идентификатор, используемый для хранения значений в памяти компьютера – _____.

Правильный ответ: переменная

Компетенции (индикаторы) ОПК-14

3. Структура данных, позволяющая хранить множество элементов одного типа, к которым можно обратиться по индексу – _____.

Правильный ответ: массив

Компетенции (индикаторы) ОПК-14

4. Переменная, которая хранит адрес памяти – _____.

Правильный ответ: указатель

Компетенции (индикаторы) ОПК-14

5. Механизм объектно-ориентированного программирования, позволяющий создавать новые классы на основе существующих, для упрощения повторного использования кода и его расширения называется _____.

Правильный ответ: наследование

Компетенции (индикаторы) ОПК-14

6. Процесс, когда функция вызывает саму себя – _____.

Правильный ответ: рекурсия

Компетенции (индикаторы) ОПК-14

7. Экземпляр класса, который имеет свои собственные значения свойств и может вызывать методы, определенные в классе – _____.

Правильный ответ: объект

Компетенции (индикаторы) ОПК-14

8. Специальная функция в классе, которая автоматически вызывается при создании нового объекта этого класса – _____.

Правильный ответ: конструктор

Компетенции (индикаторы) ОПК-14

Задания открытого типа с кратким свободным ответом

Напишите пропущенное слово (словосочетание)

1. Дополните программу на C++, которая проверяет, является ли введенное пользователем число четным или нечетным. Используйте пробелы для отделения операторов от операндов.

```
int number;
cin >> number;
if ( _____ ) {
    cout << "нечетное" << endl;
} else {
    cout << "четное" << endl;
}
```

Правильный ответ: `number % 2 != 0` / `number % 2`

Компетенции (индикаторы) ОПК-14

2. Дополните программу на C++, которая меняет значения двух переменных (a, b) местами, используя новую переменную (tmp). Используйте пробелы для отделения операторов от операндов. Вся последовательность преобразований представьте одной строкой.

```
int a, b;
cin >> a >> b;
int tmp;
```

```
_____
cout << a << b;
```

Правильный ответ: `tmp = a; a = b; b = tmp; / tmp = b; b = a; a = tmp;`

Компетенции (индикаторы) ОПК-14

3. Дополните программу на C++, которая выводит все натуральные числа от 1 до 10. Используйте пробелы для отделения операторов от операндов. Обращайте внимание на уже написанные точки с запятыми.

```
int main() {
    for (int i = 1; _____; ++i) {
        cout << i << endl;
    }
    return 0;
}
```

Правильный ответ: $i \leq 10 / i < 11$

Компетенции (индикаторы) ОПК-14

4. Дополните программу на C++, которая вычисляет факториал числа. Используйте пробелы для отделения операторов от операндов. Обращайте внимание на уже написанные точки с запятыми.

```
int main() {
    int n;
    cout << "Введите число: ";
    cin >> n;

    int factorial = 1;
    for (int i = 1; i <= n; ++i) {
        _____;
    }
    cout << "Факториал " << n << " равен: " << factorial << endl;
    return 0;
}
```

Правильный ответ: $factorial *= i / factorial = factorial * i / factorial = i * factorial$

Компетенции (индикаторы) ОПК-14

5. Дополните рекурсивную функцию C++, которая вычисляет факториал натурального числа. Используйте пробелы для отделения операторов от операндов. Обращайте внимание на уже написанные точки с запятыми.

```
int factorial(int n) {
    if (n == 1) {
        return 1;
    }
    return _____;
}
```

Правильный ответ: $n * \text{factorial}(n - 1) / \text{factorial}(n - 1) * n$

Компетенции (индикаторы) ОПК-14

6. Дополните программу на C++, размещающую 10 целых чисел пользователя в динамической памяти. Используйте пробелы для отделения операторов от операндов. Обращайте внимание на уже написанные точки с запятыми.

```
int main() {
    _____ = new int[10];
    cout << "Введите " << 10 << " целых чисел: ";
    for (int i = 0; i < 10; ++i) {
        cin >> array[i];
    }
    cout << "Вы ввели: " << endl;
    for (int i = 0; i < 10; ++i) {
        cout << array[i] << " ";
    }
    cout << endl;
    delete[] array;
    return 0;
}
```

Правильный ответ: $\text{int}^* \text{array} / \text{int}^* \text{array} / \text{int}^* \text{array} / \text{int}^* \text{array}$

Компетенции (индикаторы) ОПК-14

7. Дополните программу, реализующую класс очереди на C++. Конструктор класса должен указать, что указатель `queue` не ссылается на память. Используйте пробелы для отделения операторов от операндов. Обращайте внимание на уже написанные точки с запятыми.

```
class Queue {
private:
    int* queue; // защищаемый массив для хранения очереди
public:
    Queue() {
        _____;
    }
    ~Queue() {
        delete[] queue;
    }
};
```

Правильный ответ: `queue = NULL` / `queue = nullptr` / `queue = 0`

Компетенции (индикаторы) ОПК-14

8. Дополните программу, реализующую полиморфизм на C++. Используйте пробелы для отделения операторов от операндов.

```
class Person {
public:
    virtual void name() = 0;
    void claccName(){ cout << "claccName Person\n"; }
};

class Student : public Person{
public:
    _____{ cout << "Student метод name\n"; }
    void claccName(){ cout << "claccName Student\n"; }
};
```

Правильный ответ: `virtual void name()` / `void name() override`

Компетенции (индикаторы) ОПК-14

Задания открытого типа с развернутым ответом

1. Напишите программу на языке C++, реализующую поиск наибольшего значения в массиве из целых чисел размером 10 элементов.

Привести полный код программы.

Время выполнения – 20 мин

Ожидаемый результат:

```
#include <iostream>
using namespace std;

int main() {
    int a[10];
    for (int i = 0; i < 10; i++) {
        cin >> a[i];
    }

    int maxIndex = 0;
    for (int i = 1; i < 10; i++) {
        if (a[i] > a[maxIndex]) {
            maxIndex = i;
        }
    }

    cout << "Индекс элемента: " << maxIndex << endl;
    return 0;
}
```

Критерии оценивания:

- в программе подключены необходимые библиотеки;
- создана точка входа в программу;
- объявлен массив нужного размера и типа;
- обработка массива производится одним из видов циклов;
- программа каким-либо способом сообщает или возвращает ответ.

Компетенции (индикаторы) ОПК-14

2. Напишите программу на языке C++, которая подсчитывает среднее значение из 5 введенных чисел.

Привести полный код программы.

Время выполнения – 20 мин

Ожидаемый результат:

```
#include <iostream>
using namespace std;

int main() {
    // Инициализация массива для хранения 5 чисел
    double number;
    double sum = 0.0;

    cout << "Введите 5 чисел: ";
    for (int i = 0; i < 5; i++) {
        cin >> number;
        sum += number;
    }
    cout << "Среднее значение: " << sum / 5 << endl;
    return 0;
}
```

Критерии оценивания:

- в программе подключены необходимые библиотеки;
- создана точка входа в программу;
- объявлен массив нужного размера и типа при необходимости;
- повышает балл решение без использования массива;
- при расчете учтено, что ответ может быть не целым числом;
- программа каким-либо способом сообщает или возвращает ответ.

Компетенции (индикаторы) ОПК-14

3. Напишите программу на языке C++, для сортировки массива по возрастанию методами пузырька.

Привести полный код программы.

Время выполнения – 20 мин

Ожидаемый результат:

```
#include <iostream>
using namespace std;

int main() {
    const int size = 7;
    int arr[size] = {6, 4, 5, 1, 2, 7, 9};
    for (int i = 0; i < size - 1; i++) {
        for (int j = 0; j < size - i - 1; j++) {
            if (arr[j] > arr[j + 1]) {
                int tmp = arr[j];
                arr[j] = arr[j + 1];
                arr[j + 1] = tmp;
            }
        }
    }
    for (int i = 0; i < size; i++) {
        cout << a[i];
    }
    return 0;
}
```

Критерии оценивания:

- в программе подключены необходимые библиотеки;
- создана точка входа в программу;
- объявлен массив нужного размера и типа;
- применены вложенные циклы;
- программа каким-либо способом сообщает или возвращает ответ.

Компетенции (индикаторы) ОПК-14

4. Напишите программу на языке C++, вычисляющую минимальное число большее 200, которое нацело делится на 17.

Привести полный код программы.

Время выполнения – 20 мин

Ожидаемый результат:

```
#include <iostream>
using namespace std;

int main() {
    int x = 201;
    while (x % 17) {
        x++;
    }
    cout << "Ответ: " << x << endl;
    return 0;
}
```

Критерии оценивания:

- в программе подключены необходимые библиотеки;
- создана точка входа в программу;
- правильно применен цикл;
- программа каким-либо способом сообщает или возвращает ответ.

Компетенции (индикаторы) ОПК-14

5. Напишите программу на языке C++, содержащую функцию, которая по координатам точки (x, y) вернет четверть декартовых координат. Дополнительное условие: $x \neq 0$ и $y \neq 0$.

Привести полный код программы.

Время выполнения – 45 мин

Ожидаемый результат:

```
#include <iostream>
using namespace std;

int determineQuadrant(int x, int y) {
    if (x == 0 || y == 0) {
        return -1;
    }

    if (x > 0) {
```

```

        if (y > 0) {
            return 1;
        } else {
            return 4;
        }
    } else {
        if (y > 0) {
            return 2;
        } else {
            return 3;
        }
    }
}

int main() {
    int x, y;

    cout << "Введите координаты точки (x, y): ";
    cin >> x >> y;

    int quadrant = determineQuadrant(x, y);
    if (quadrant < 0) {
        cout << "ошибка" << endl;
        return 1;
    }
    cout << "Точка в " << quadrant << "-й четверти." << endl;
    return 0;
}

```

Критерии оценивания:

- в программе подключены необходимые библиотеки;
- создана точка входа в программу;
- создана функция по заданию;
- программа каким-либо способом сообщает или возвращает ответ.

Компетенции (индикаторы) ОПК-14

6. Напишите программу на языке C++, которая создаст массив из 20 случайных целых чисел из отрезка [0;13], выведет его на экран, создаст второй массив нужного размера только из чётных элементов первого массива, если они там есть, и выведет его на экран.

Привести полный код программы.

Время выполнения – 45 мин

Ожидаемый результат:

```

#include <iostream>
using namespace std;

int main() {
    const int size = 20;
    int firstArray[size];
    for (int i = 0; i < size; ++i) {
        firstArray[i] = rand() % 14;
    }

    cout << "Первый массив: ";
    for (int i = 0; i < size; ++i) {
        cout << firstArray[i] << " ";
    }
    cout << endl;

    int evenCount = 0;
    for (int i = 0; i < size; ++i) {
        if (firstArray[i] % 2 == 0) {
            ++evenCount;
        }
    }

    if (evenCount > 0) {
        int *secondArray = new int[evenCount];
        for (int i = 0, i2 = 0; i < size; ++i) {
            if (firstArray[i] % 2 == 0) {
                secondArray[i2++] = firstArray[i];
            }
        }
        cout << "Второй массив: ";
        for (int i = 0; i < evenCount; ++i) {
            cout << secondArray[i] << " ";
        }
        cout << endl;
        delete[] secondArray;
    } else {
        cout << "Четных элементов в первом массиве нет." << endl;
    }
    return 0;
}

```

Критерии оценивания:

- в программе подключены необходимые библиотеки;
- создана точка входа в программу;
- объявлен массив нужного размера и типа;
- размер второго массива получен, а память выделена динамически;
- программа каким-либо способом сообщает или возвращает ответ.

Компетенции (индикаторы) ОПК-14

7. Напишите программу на языке C++, реализующую очередь через односвязный список.

Привести полный код программы.

Время выполнения – 60 мин

Ожидаемый результат:

```
#include <iostream>
using namespace std;

struct Node {
    int data;
    Node* next;
    Node(int value) {
        data = value;
        next = nullptr;
    }
    ~Node() {
        if (next != nullptr) {
            delete next;
        }
    }
};

class Queue {
private:
    Node* head;
    Node* tail;
public:
    Queue() {
        head = nullptr;
        tail = nullptr;
    }

    ~Queue() {
        if (head != nullptr) {
            delete head;
        }
    }

    void enqueue(int value) {
        if (head == nullptr) {
            head = tail = new Node(value);
        } else {
            tail->next = new Node(value);
            tail = tail->next;
        }
    }
};
```

```

    }

    int dequeue() {
        int value = head->data;
        Node* temp = head;
        head = head->next;
        temp = nullptr;
        delete temp;
        if (head == nullptr) {
            tail = nullptr;
        }
        return value;
    }

    bool isEmpty() const {
        return head == nullptr;
    }
};

int main() {
    Queue queue;
    queue.enqueue(3);
    queue.enqueue(5);
    queue.enqueue(7);
    cout << "Извлеченный элемент: " << queue.dequeue() << endl;
    cout << "Очередь пуста? " << (queue.isEmpty() ? "Да" : "Нет")
<< endl;
    return 0;
}

```

Критерии оценивания:

- в программе подключены необходимые библиотеки;
- создана точка входа в программу;
- объявлены классы для организации очереди;
- в классах объявлены необходимые указатели для организации очереди;
- реализован метод добавления элемента в очередь;
- реализован метод обработки элемента из очереди;
- динамически созданные элементы удаляются в конце использования;
- программа каким-либо способом сообщает или возвращает ответ.

Компетенции (индикаторы) ОПК-14

8. Напишите программу на языке C++, содержащую шаблонную функцию которая возвращает указатель на наибольший элемент в массиве.

Привести полный код программы.

Время выполнения – 45 мин

Ожидаемый результат:

```
#include <iostream>
using namespace std;

template <typename T>
T* maxElement(T* array, int size) {
    if (size <= 0) {
        return nullptr;
    }

    T* maxPtr = array;
    for (int i = 1; i < size; ++i) {
        if (array[i] > *maxPtr) {
            maxPtr = array + i;
        }
    }
    return maxPtr;
}

int main() {
    int intArray[] = {1, 22, 3, 44, 5};
    int sizeInt = sizeof(intArray) / sizeof(int);
    int* maxIntPtr = maxElement(intArray, sizeInt);
    cout << "Наибольший из int: " << *maxIntPtr << endl;

    double doubleArray[] = {0.1, 2.4, 0.4, 33.1};
    int sizeDouble = sizeof(doubleArray) / sizeof(double);
    double* maxDoublePtr = maxElement(doubleArray, sizeDouble);
    cout << "Наибольший из double: " << *maxDoublePtr << endl;
    return 0;
}
```

Критерии оценивания:

- в программе подключены необходимые библиотеки;
- создана точка входа в программу;
- создана шаблонная функция и правильно применен шаблонный тип;
- программа каким-либо способом сообщает или возвращает результат.

Компетенции (индикаторы) ОПК-14

Экспертное заключение

Представленный комплект оценочных материалов по дисциплине «Программирование и основы алгоритмизации систем управления» соответствует требованиям ФГОС ВО.

Предлагаемые оценочные материалы адекватны целям и задачам реализации основной профессиональной образовательной программы по направлению подготовки 15.03.04 Автоматизация технологических процессов и производств.

Виды оценочных средств, включенные в представленный фонд, отвечают основным принципам формирования ФОС.

Разработанные и представленные для экспертизы оценочные материалы рекомендуются к использованию в процессе подготовки обучающихся по указанному направлению.

Председатель учебно-методической
комиссии института компьютерных
систем и информационных
технологий



Ветрова Н. Н.

Лист изменений и дополнений

| № п/п | Виды дополнений и изменений | Дата и номер протокола заседания кафедры (кафедр), на котором были рассмотрены и одобрены изменения и дополнения | Подпись (с расшифровкой) заведующего кафедрой (заведующих кафедрами) |
|----------|--|--|---|
| 1 | В фонд оценочных средств добавлен комплект оценочных материалов | 25.02.2025 г., №14 |  А.В. Колесников |
| | | | |
| | | | |
| | | | |