

Комплект оценочных материалов по дисциплине
«Технология программирования процессов автоматического
управления»

Задания закрытого типа

Задания закрытого типа на выбор правильного ответа

Выберите один правильный ответ

1. Что такое многопоточное приложение?

А) приложение, которое работает только в одном потоке

Б) приложение, которое использует несколько потоков для выполнения задач одновременно

В) приложение, которое работает только в реальном времени

Г) приложение, которое использует только один процессор

Правильный ответ: Б

Компетенции (индикаторы): ПК-1

2. Какой из следующих объектов используется для управления потоками в многопоточных приложениях?

А) мьютексы

Б) байт-коды

В) массивы

Г) циклы

Правильный ответ: А

Компетенции (индикаторы): ПК-1

3. Что такое критическая секция в контексте многопоточных приложений?

- А) механизм ускорения выполнения потоков за счет усложненных секций
- Б) механизм для предотвращения одновременного доступа к защищаемому ресурсу несколькими потоками
- В) механизм для разделения потоков на подпотоки
- Г) механизм для объединения потоков

Правильный ответ: Б

Компетенции (индикаторы): ПК-1

4. Что такое deadlock (смертельный захват) в многопоточных приложениях?

- А) ситуация, когда все потоки выполняются одновременно
- Б) ситуация, когда потоки не могут продолжить выполнение из-за ожидания ресурсов друг у друга
- В) ситуация, когда один поток завершает работу всех остальных
- Г) ситуация, когда потоки работают без блокировок

правильный ответ: Б

Компетенции (индикаторы): ПК-1

Задания закрытого типа на установление соответствия

Установите правильное соответствие.

Каждому элементу левого столбца соответствует только один элемент правого столбца.

1. Установите соответствие между методом управления потоками и его описанием в многопоточных приложениях:

Описание метода управления	Метод управления
1) механизм для блокировки ресурса до его освобождения	А) семафоры
2) механизм для синхронизации доступа к ресурсу с помощью счетчика	Б) мьютексы
3) высокоуровневый механизм синхронизации, включающий условия и мьютексы	В) мониторы
4) Механизм для уведомления потоков о событиях	Г) события

Правильный ответ

1	2	3	4
Б	А	В	Г

Компетенции (индикаторы): ПК-1

2. Установите соответствие между сетевым протоколом и его назначением:

Назначение сетевого протокола	Сетевой протокол
1) безопасное удаленное управление компьютерами и передачу данных	А) HTTPS
2) передача файлов	Б) FTP
3) быстрая передача данных без установления соединения	В) SHH
4) безопасная передача данных между веб-серверами и клиентами	Г) UDP

Правильный ответ

1	2	3	4
В	Б	Г	А

Компетенции (индикаторы): ПК-1

3. Установите соответствие между типом сокета и его назначением в сетевых приложениях:

Назначение сокета	Тип сокета
1) обеспечивает упорядоченную доставку данных без установления соединения	А) потоковый сокет (Stream Socket)
2) позволяет работать с низкоуровневыми протоколами	Б) датиграммный сокет (Datagram Socket)
3) используется для передачи данных без установления соединения	В) сырой сокет (Raw Socket)
4) используется для надежной передачи данных с установлением соединения	Г) сеансовый сокет (Sequence Packet Socket)

Правильный ответ

1	2	3	4
Г	В	Б	А

Компетенции (индикаторы): ПК-1

4. Установите соответствие между функцией WinSock и ее назначением в сетевых приложениях:

Назначение функции	Название функции
1) функция для принятия входящего соединения	А) bind
2) функция для привязки сокета к адресу и порту	Б) listen
3) функция для установки сокета в режим прослушивания входящих соединений	В) accept
4) функция для отправки данных через сокет	Г) send
5) функция для получения данных через сокет	Д) recv

Правильный ответ

1	2	3	4	5
В	А	Б	Г	Д

Компетенции (индикаторы): ПК-1

Задания закрытого типа на установление правильной последовательности

Установите правильную последовательность.

Запишите правильную последовательность букв слева направо.

1. Установите правильную последовательность действий при работе с критическими секциями на C++:

- А) выполнить защищаемые операции
- Б) инициализировать мьютекс (критическую секцию)
- В) заблокировать мьютекс (критическую секцию)
- Г) разблокировать мьютекс (критическую секцию)

Правильный ответ: Б, В, А, Г

Компетенции (индикаторы): ПК-1

2. Установите правильную последовательность действий при создании сервера на сокетах C++:

- А) `listen(listenSocket, SOMAXCONN);`
- Б) `bind(listenSocket, (SOCKADDR*)&service, sizeof(service));`
- В) `accept(listenSocket, NULL, NULL)`
- Г) `recv(clientSocket, recvBuf, sizeof(recvBuf), 0);`

Правильный ответ: Б, А, В, Г

Компетенции (индикаторы): ПК-1

3. Расположите следующие блоки кода WinApi C++ в порядке увеличения времени ожидания. Исходные данные:

```
DWORD WINAPI f(PVOID param) {  
    while (true) {  
        cout << "+" ;  
    }  
}  
HANDLE h0 = NULL ;  
HANDLE h1 = CreateThread(NULL, 0, f, NULL, 0, NULL) ;  
HANDLE h2 = CreateThread(NULL, 0, f, NULL, 0, NULL) ;  
HANDLE h3 = CreateThread(NULL, 0, f, NULL, 0, NULL) ;
```

А) WaitForSingleObject (h0, INFINITE);

Б) WaitForSingleObject (h1, INFINITE);

В) WaitForSingleObject (h2, 10);

Г) WaitForSingleObject (h3, 100);

Правильный ответ: А, В, Г, Б

Компетенции (индикаторы): ПК-1

4. Расположите протоколы по снижению уровня надежности и защищенности:

А) HTTP

Б) HTTPS

В) UDP

Правильный ответ: Б, А, В

Компетенции (индикаторы): ПК-1

Задания открытого типа

Задания открытого типа на дополнение

Напишите пропущенное слово (словосочетание)

1. Протокол _____ часто используется для надежной передачи данных в сетевых приложениях.

Правильный ответ: TCP

Компетенции (индикаторы): ПК-1

2. Протокол _____ часто используется для передачи данных в сетевых приложениях, где важна скорость и не требуется высокая надежность.

Правильный ответ: UDP

Компетенции (индикаторы): ПК-1

3. Дополните строку программы на WinApi C++ ключевым словом.

_____ (hEvent, INFINITE); // ожидание события

Правильный ответ: WaitForSingleObject

Компетенции (индикаторы): ПК-1

4. Объект (логический канал связи), который представляет собой соединение между клиентом и сервером и используется для передачи данных в обоих направлениях — _____.

Правильный ответ: сокет

Компетенции (индикаторы): ПК-1

Задания открытого типа с кратким свободным ответом

Напишите пропущенное слово (словосочетание)

1. Дополните функцию на языке C++, чтобы она могла быть вызвана

```
HANDLE h = CreateThread(NULL, 0, ThreadProc, hEvent, 0, NULL);
```

Используйте пробелы для отделения операторов от операндов.

```
DWORD WINAPI ThreadProc(_____) {  
    HANDLE hEvent = (HANDLE)param;  
    WaitForSingleObject(hEvent, INFINITE);  
    cout << "Событие получено" << endl;  
    return 0;  
}
```

Правильный ответ: LPVOID param / PVOID param / void *param / void* param

Компетенции (индикаторы): ПК-1

2. В WinAPI для работы с событиями используются несколько ключевых функций, которые позволяют создавать, управлять и синхронизировать потоки. Укажите название функции, если она имеет вид:

```
BOOL _____(  
    HANDLE hEvent  
);
```

Правильный ответ: SetEvent / ResetEvent / CloseHandle

Компетенции (индикаторы): ПК-1

3. В многопоточных приложениях объект пользовательского режима «критическая секция» и объект ядра «_____» гарантируют потокам взаимоисключающий доступ к единственному ресурсу.

Правильный ответ: мьютекс / mutex / объект мьютекса

Компетенции (индикаторы): ПК-1

4. При программировании сетевых приложений на C++, функция _____ используется для отправки данных через сокет.

Правильный ответ: send / sendto

Компетенции (индикаторы): ПК-1

Задания открытого типа с развернутым ответом

1. Объясните процесс создания и завершения потоков в Windows API. Укажите основные компоненты, которые включает в себя поток при создании. Опишите параметры функции `CreateThread` и их назначение. Какими способами можно завершить поток, и какие особенности у каждого из них? Рассмотрите ситуацию, когда необходимо экстренно завершить поток. Какие последствия это может вызвать, и как правильно обработать такое завершение, минимизировав возможные проблемы?

Время выполнения – 30 мин

Критерии оценивания:

- описана последовательность работы с функциями потоков;
- дано описание структуры потока;
- разбор каждого параметра `CreateThread`;
- дан сравнительный анализ способов завершения потока;
- описаны основные ситуации, к которым может привести экстренное завершение потока.

Компетенции (индикаторы): ПК-1, ПК-4

2. Дайте максимально развернутый ответ с примерами. Как обеспечить безопасную работу с разделяемыми ресурсами в многопоточном приложении на языке C++?

Время выполнения – 30 мин

Критерии оценивания:

- дано описание критических секций, их назначения и примеры использования;
- дано описание основных проблем, которые могут возникнуть при работе с разделяемыми ресурсами без использования критических секций;

- описана вероятность возникновения «смертельных захватов» (deadlock) и показаны способы их избежать;

- показаны преимущества и недостатки использования критических секций по сравнению с другими механизмами синхронизации.

Компетенции (индикаторы): ПК-1, ПК-4

3. Напишите основную часть многопоточной программы на C++, например, с использованием Windows API, реализующей алгоритм работы нескольких исполнительных устройств по алгоритму: $\Rightarrow 1,2,3 \Leftarrow (2,3),1$.

Время выполнения – 45 мин

Критерии оценивания:

- в программе подключены необходимые библиотеки;
- создана точка входа в программу;
- если в ходе решения созданы динамически созданные элементы, то они удаляются в конце использования;
- синхронизация синхронизация между потоками выполнена корректно;
- в реализации отсутствуют «смертельные захваты» (deadlock);
- программа каким-либо способом сообщает или возвращает ответ.

Компетенции (индикаторы): ПК-1, ПК-4

4. Напишите программу на языке C++, реализующую многопоточный TCP сервер на WinSock.

Привести код основных участков программы.

Время выполнения – 60 мин

Ожидаемый результат:

```
#include <iostream>
#include <string>
#include <Winsock2.h>
using namespace std;

#pragma comment(lib, "Ws2_32.lib")
```

```

void ClientHandler(SOCKET clientSocket) {
    char buffer[1024];
    while (true) {
        int bytesRead = recv(clientSocket, buffer,
sizeof(buffer), 0);
        if (bytesRead <= 0) break;
        buffer[bytesRead] = 0;
        cout << "Получено от клиента: " << buffer << endl;
        strcpy(buffer, "Сервер получил запрос ");
        strcat(buffer, itoa(bytesRead));
        send(clientSocket, buffer, strlen(buffer), 0);
    }
    closesocket(clientSocket);
}

DWORD WINAPI ClientHandlerThread(LPVOID param) {
    SOCKET clientSocket = (SOCKET) param;
    ClientHandler(clientSocket);
    return 0;
}

int main() {
    WSADATA wsaData;
    if (WSAStartup(MAKEWORD(2, 2), &wsaData) != 0) {
        cout << "Ошибка инициализации Winsock" << endl;
        return 1;
    }

    SOCKET serverSocket = socket(AF_INET, SOCK_STREAM,
IPPROTO_TCP);
    if (serverSocket == INVALID_SOCKET) {
        cout << "Ошибка создания сокета" << endl;
        WSACleanup();
        return 1;
    }

    sockaddr_in serverAddress;
    serverAddress.sin_family = AF_INET;
    serverAddress.sin_addr.s_addr = INADDR_ANY;
    serverAddress.sin_port = htons(8080);

    if (bind(serverSocket, (sockaddr*)&serverAddress,
sizeof(serverAddress)) == SOCKET_ERROR) {
        cout << "Ошибка привязки" << endl;
        closesocket(serverSocket);
        WSACleanup();
        return 1;
    }

    if (listen(serverSocket, SOMAXCONN) == SOCKET_ERROR) {
        cout << "Ошибка прослушивания" << endl;
        closesocket(serverSocket);
    }
}

```

```

        WSACleanup();
        return 1;
    }

    cout << "Сервер запущен на порту 8080" << endl;

    while (true) {
        SOCKET clientSocket = accept(serverSocket, NULL, NULL);
        if (clientSocket != INVALID_SOCKET) {
            CreateThread(NULL, 0, ClientHandlerThread, (PVOID)
clientSocket, 0, NULL);
            cout << "Новое подключение: " << clientSocket <<
endl;
        }
    }
    closesocket(serverSocket);
    WSACleanup();
    return 0;
}

```

Критерии оценивания:

- в программе подключены необходимые библиотеки;
- создана точка входа в программу;
- предусмотрен алгоритм запуска и остановки работы работы с сетью;
- прописана последовательность создания сервера;
- создана и вызвана функция поток, обрабатывающая принятого клиента;
- программа каким-либо спообом сообщает или возвращает результат работы с клиантами.

Компетенции (индикаторы): ПК-1

Экспертное заключение

Представленный комплект оценочных материалов по дисциплине «Технология программирования процессов автоматического 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