Правила проведения зачета

по дисциплине **«Параллельное программирование»**

1. Зачет по дисциплине «Параллельное программирование» в 2019 году пройдет в письменной форме, 18 декабря (в среду (во время лекции)), результаты и подведение итогов 26 декабря (в четверг (во время лабораторной)).

2. Билет к зачету будет содержать два теоретических вопроса и два практических задания (задачи).

3. К выполнению задания допускаются все студенты без исключения. Зачет получают студенты успешно выполнившие все виды работ (лабораторные, РГР, зачет).

Список теоретических вопросов и типов задач

к зачету по дисциплине **«Параллельное программирование»**

Вопросы:

1. Закон Амдала.
2. Классификация Флинна с пояснениями.
3. Основные модели создания и функционирования потоков.
4. Гонка данных, взаимная блокировка, бесконечное ожидание.
5. Критическая секция, мьютексный семафор.
6. Активное/пассивное ожидание, спинлок.
7. Средства описания параллельных алгоритмов в UML 2.0
8. OpenMP: Модель выполнения программы, создание потоков.
9. OpenMP: Разделение работы между потоками. Опции, влияющие на разделение данных.
10. OpenMP: Средства синхронизации.
11. POSIX: Создание и завершение потоков.
12. POSIX: Средства синхронизации.
13. C#: Создание и завершение потоков.
14. C#: Средства синхронизации.
15. C++: Виды атомарных объектов и виды операций над ними (методы).
16. C++: `std::memory_order`: Последовательно согласованная модель.
17. C++: `std::memory_order`: Модель захвата-освобождения.
18. C++: `std::memory_order`: Модель потребления.
19. C++: `std::memory_order`: Ослабленная модель.
20. MPI: Создание и запуск MPI программ.
21. MPI: Парные взаимодействия.
22. MPI: Коллективные взаимодействия.
23. MPI: Производные типы данных.
24. CUDA: Аппаратные особенности GPU, типы памяти, возможности синхронизации.
25. CUDA: Расширение языка C для работы с GPU.

Типы задач:

1. Дан текст программы (полный или фрагмент) и описание задачи. Найти ошибку в программе. Варианты задач и средств реализации аналогичны лабораторным работам и РГР.
2. Прокомментировать заданный текст программы (полный или фрагмент). Языки программирования и используемые библиотеки аналогичны лабораторным работам и РГР.