

Создание проекта на языке DPC++ САМОСТОЯТЕЛЬНАЯ РАБОТА

<http://www.kimrt.ru>

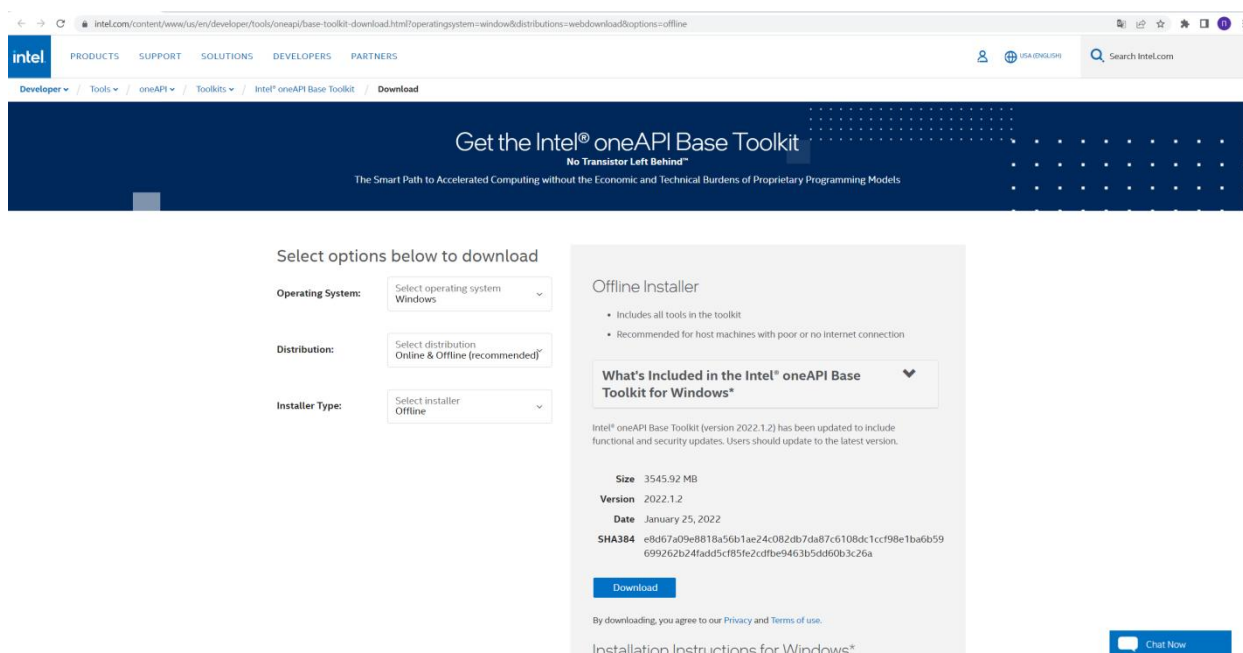
Онлайн-курс

[Суперкомпьютерные технологии в задачах моделирования](#)

DPC++ доступен в репозитории GitHub (github.com/intel/llvm). Инструкции по началу работы с DPC++, в том числе о том, как создать компилятор с открытым исходным кодом с помощью клона из GitHub, можно найти на странице intel.github.io/llvm-docs/GetStartedGuide.html. Существуют также связанные версии компилятора DPC++, с дополнительными инструментами и библиотеками для программирования и поддержки DPC++, доступные как часть более крупного проекта oneAPI. Проект обеспечивает широкую поддержку гетерогенных систем, включая библиотеки, отладчики и другие инструменты, известные как oneAPI. Инструменты oneAPI, включая DPC++, доступны бесплатно (oneapi.com/implementations). Официальную документацию по компилятору oneAPI DPC++, включая список расширений, можно найти по адресу intel.github.io/llvm-docs [1].

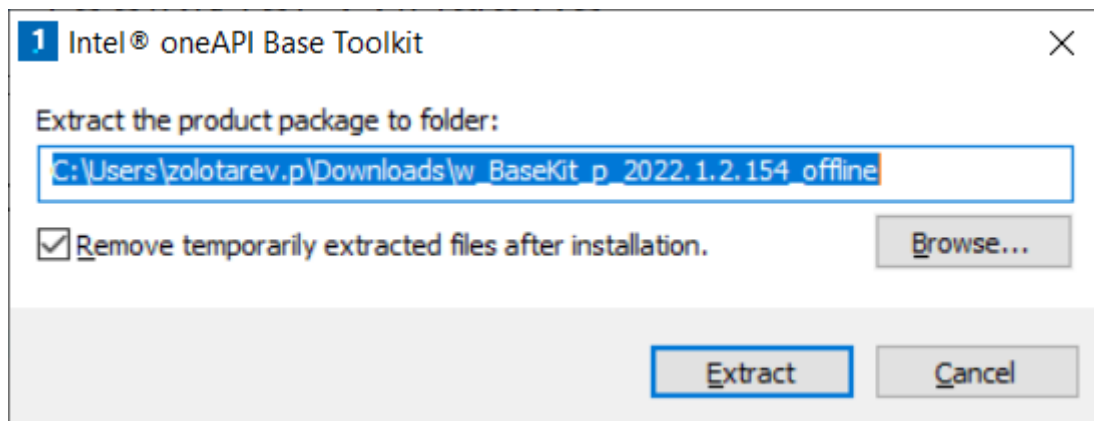
Установка oneAPI. Для загрузки файла установки перейдите по ссылке ниже. Выберите операционную систему (Windows) и тип установщика (offline), после чего нажмите кнопку загрузки файла.

<https://www.intel.com/content/www/us/en/developer/tools/oneapi/base-toolkit-download.html>



<http://www.kimrt.ru>

После загрузки файла запустите его и распакуйте его.



В следующем окне примите лицензионное соглашение и нажмите на выборочную установку (custom installation).

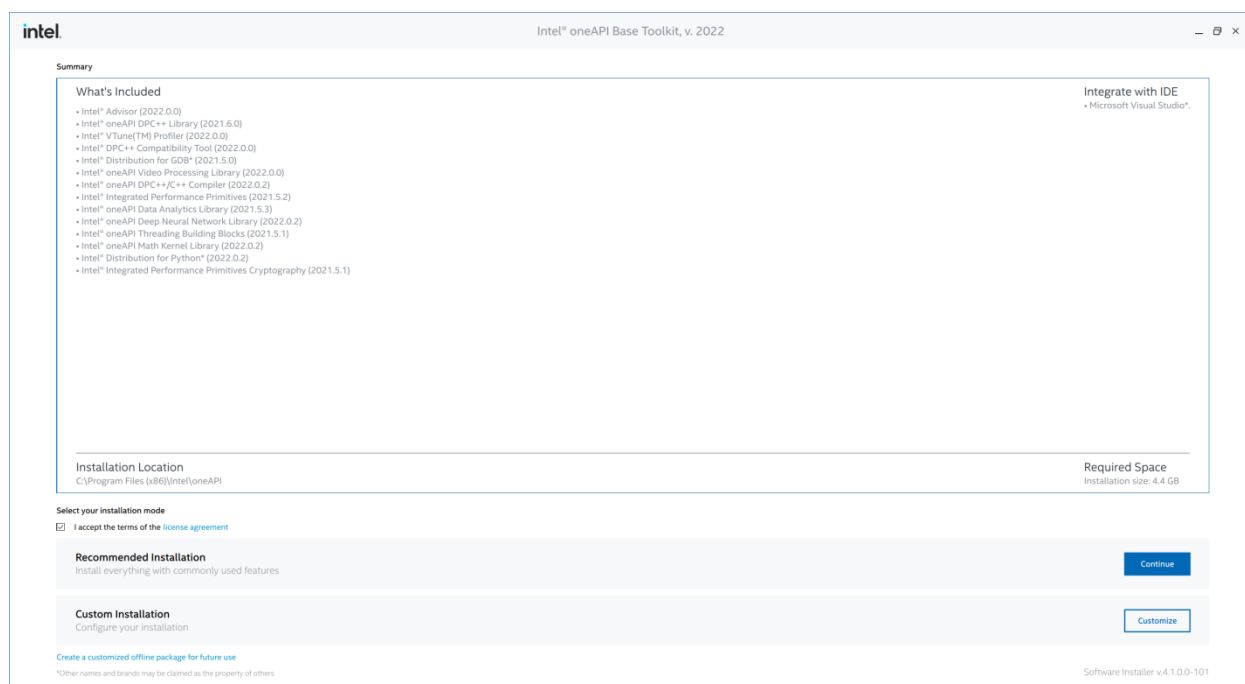
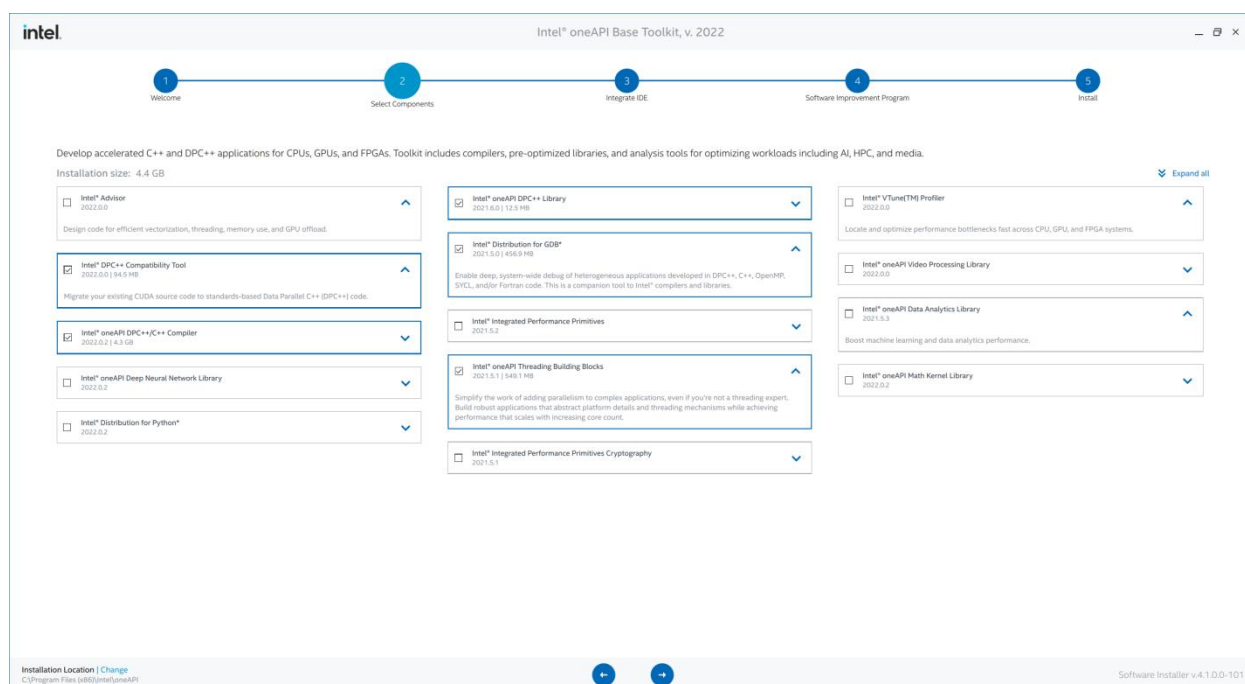


Таблица 1. Компоненты oneAPI

Компоненты	Назначение
Intel Advisor	Разработайте код для эффективной векторизации, многопоточности и разгрузки ускорителей [2].
Intel oneAPI DPC++ Library	Ускорьте параллельные рабочие нагрузки данных с помощью этих ключевых алгоритмов и функций повышения производительности.
Intel VTune Profiler	Находите и оптимизируйте узкие места производительности в системах ЦП, ГП и ПЛИС.
Intel DPC++ Compatibility Tool	Перенесите устаревший код CUDA в многоплатформенную программу в коде SYCL с помощью этого помощника.
Intel Distribution for GDB	Включите глубокую общесистемную отладку кода SYCL, C, C++ и Fortran.

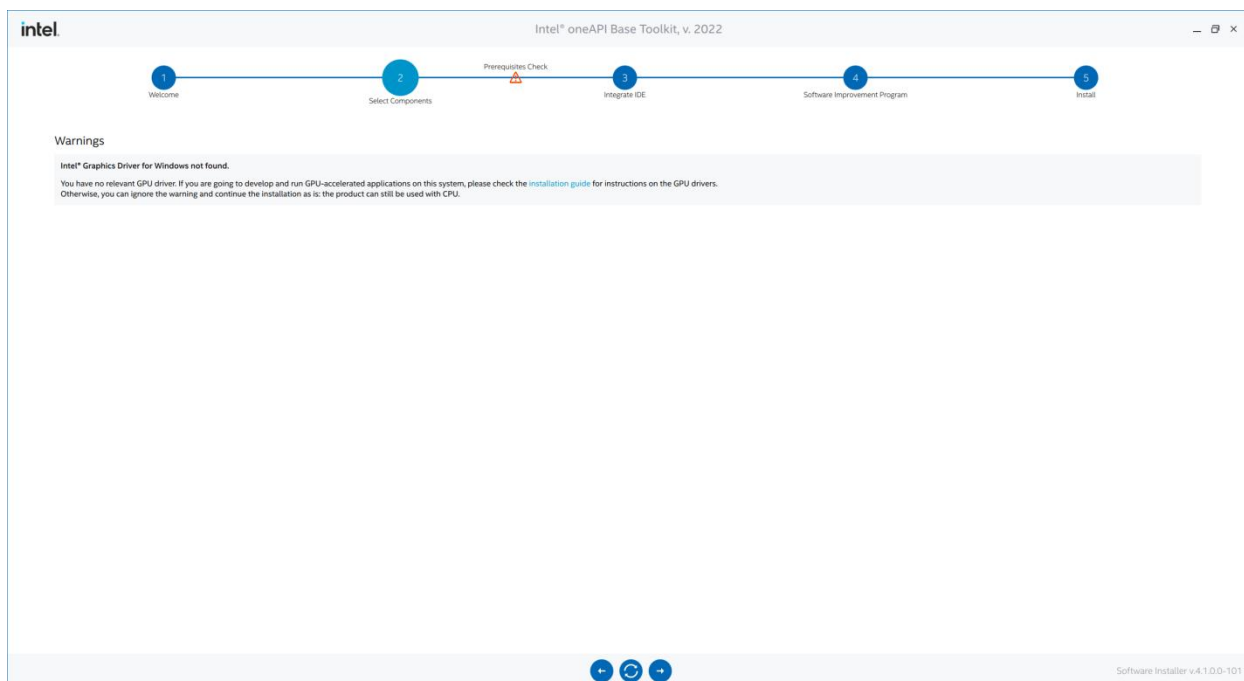
Intel oneAPI Video Processing Library	Обеспечьте быстрое высококачественное декодирование, кодирование, транскодирование и обработку видео в режиме реального времени для вещания, прямой трансляции и VOD, облачных игр и многого другого.
Intel oneAPI DPC++/C++ Compiler	Компилируйте и оптимизируйте код C++ и SYCL для целевых архитектур CPU, GPU и FPGA.

Отметьте галочкой нужные компоненты (табл. 1), например, как на скриншоте ниже.

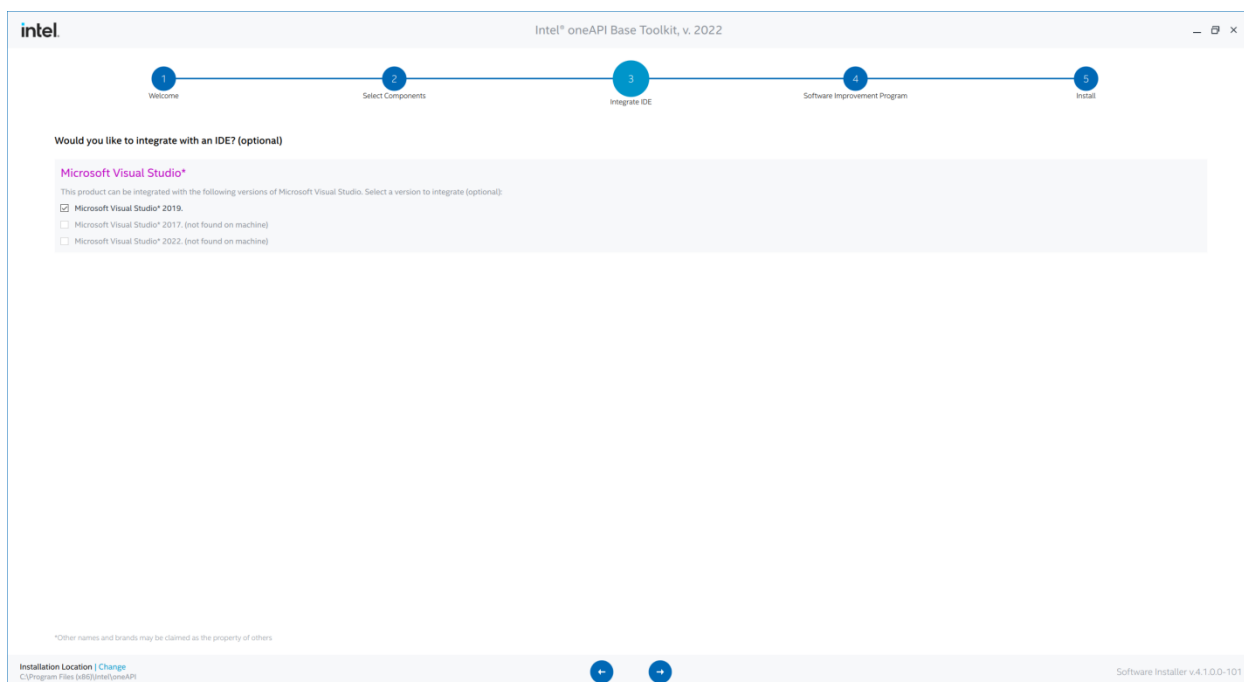


На следующем шаге может появиться предупреждение об отсутствии Intel Graphics Driver. Этот драйвер необходим для разработки программ выполняющихся на графическом процессоре Intel, поэтому пропускаем это предупреждение. При необходимости необходимый драйвер можно загрузить здесь <https://www.intel.ru/content/www/ru/ru/support/products/80939/graphics.html> или с помощью ассистента по ссылке ниже.

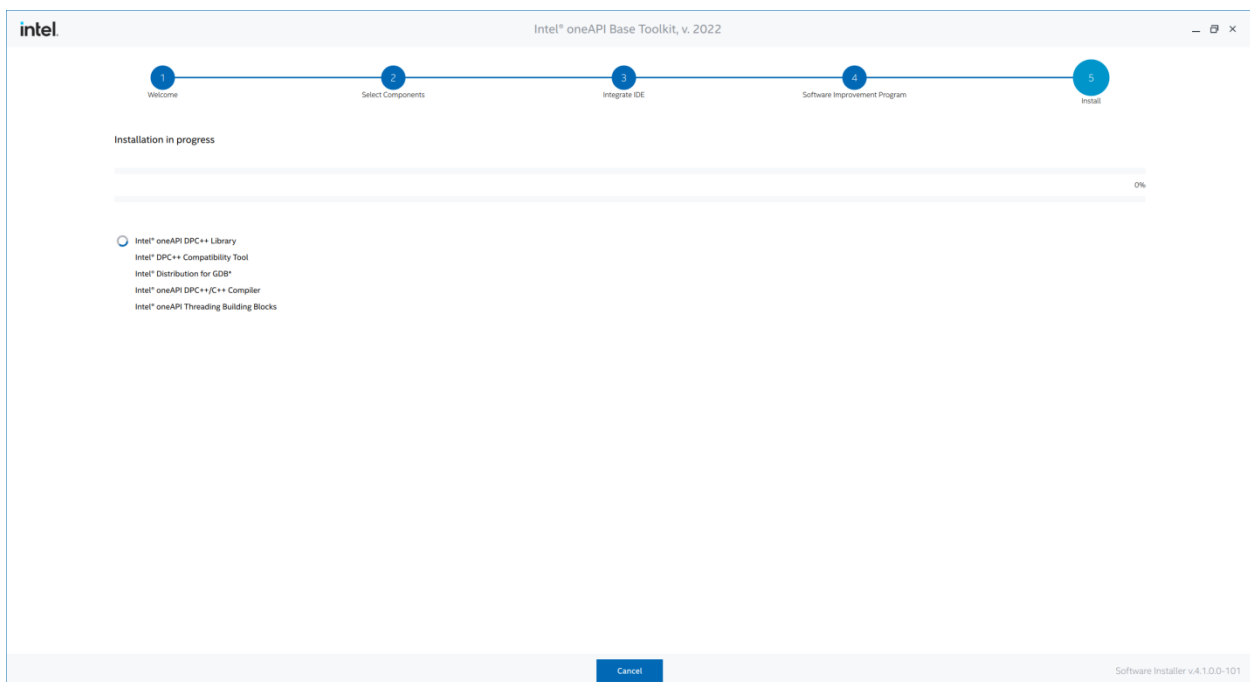
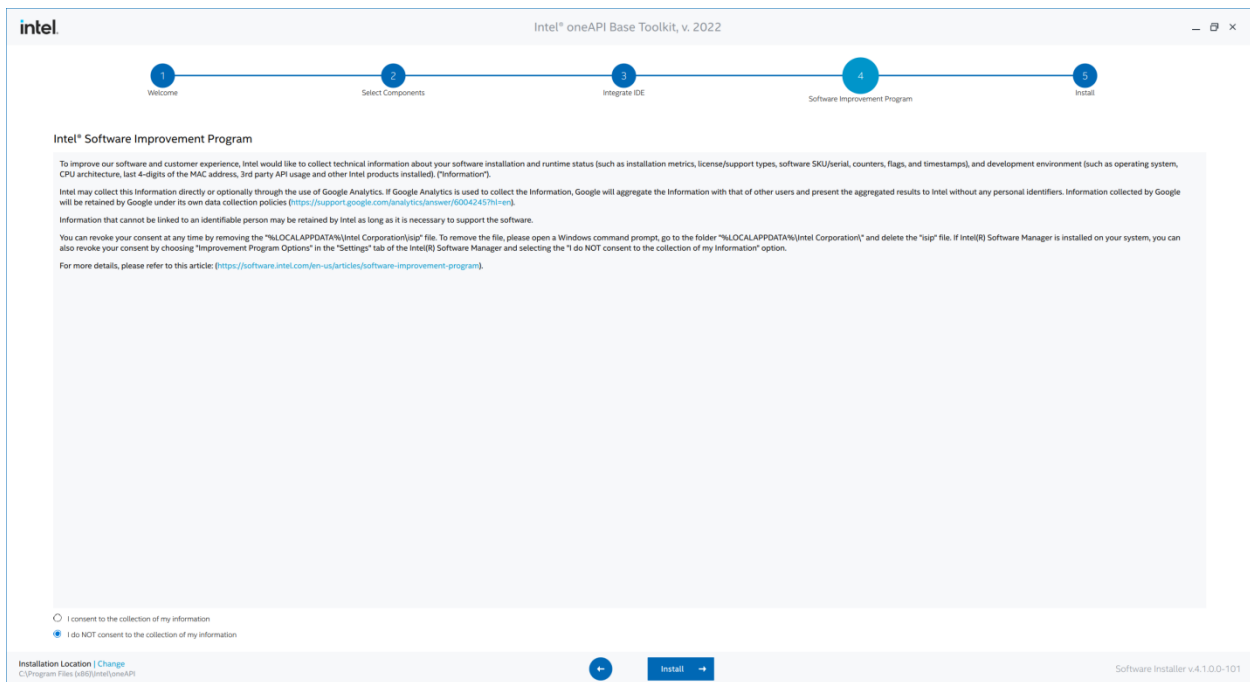
<https://www.intel.ru/content/www/ru/ru/support/detect.html>



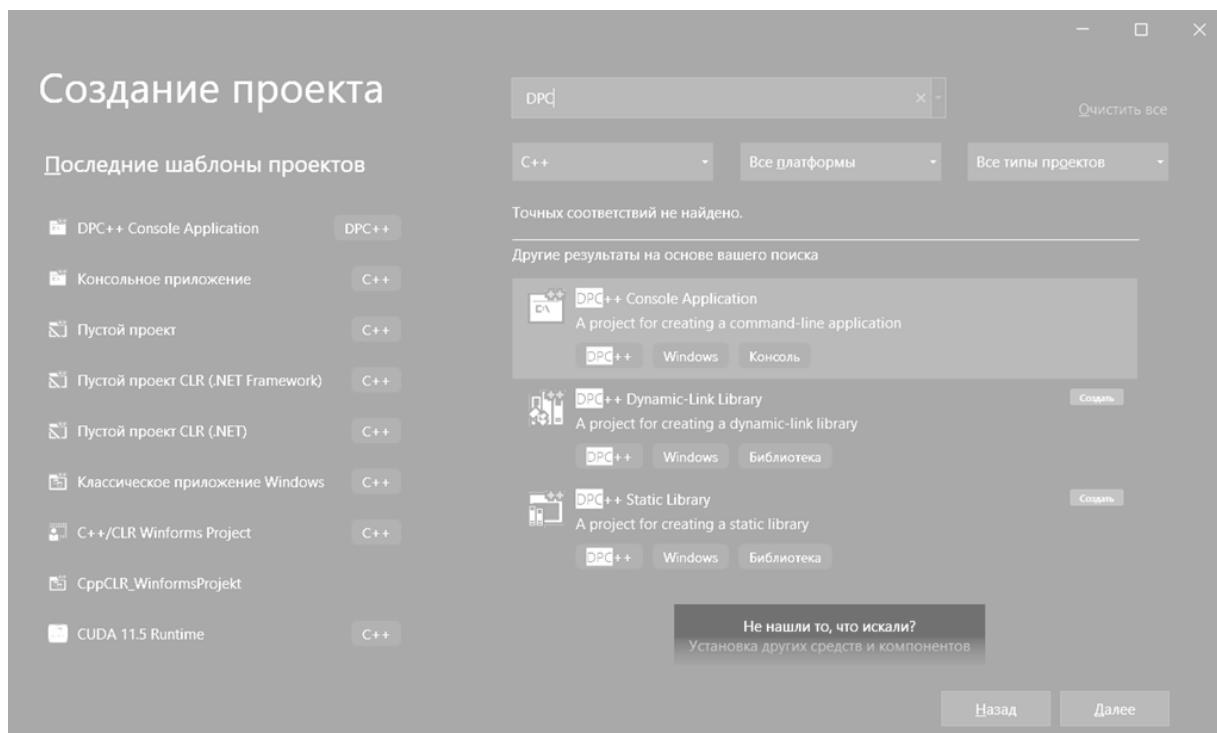
В следующем окне выбираем VS 2019 для интеграции с oneAPI.



Запускаем установку и ожидаем ее завершения.



После установки запускаем Visual Studio и создаем проект *DPC++ Console Application* [3].



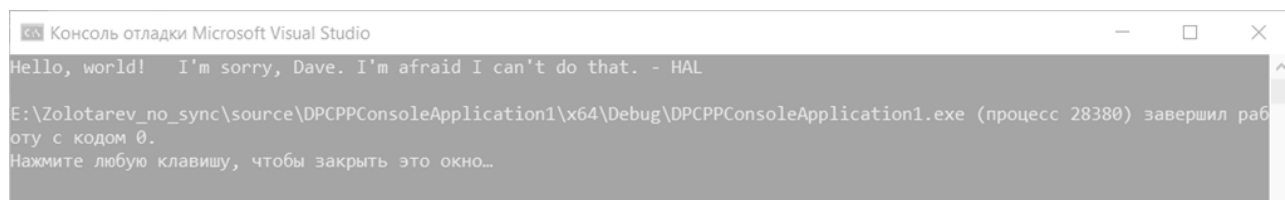
Заменяем код в сpp файле на код из листинга 1. Данная программа расшифровывает сообщение, зашифрованное шифром Цезаря со сдвигом на единицу [1].

Листинг 1. Программа расшифровки сообщения

```
#include <CL/sycl.hpp> // подключение sycl
#include <iostream>
using namespace sycl;
// строка для расшифровки
const std::string secret{
    "Ifmmp-!xpsme\"\\012J(n!tpssz-!Ebwf/!"
    "J(n!bgsbje!J!dbo(u!ep!uibu/!..IBM\\01" };
const auto sz = secret.size(); // размер строки
int main() {
    queue Q; // устанавливает очередь для рабочих запросов,
    направленных к конкретному устройству
    char* result = malloc_shared<char>(sz, Q); // создание
    распределения для данных, совместно используемых на устройстве
    std::memcpy(result, secret.data(), sz); // копирование данных
    // распределение очереди на устройстве
    Q.parallel_for(sz, [=](auto& i) {
        result[i] -= 1; // изменение кода символа
    }).wait();
    std::cout << result << "\\n"; // вывод результата
    return 0;
}
```

Собираем проект и запускаем.

<http://www.kimrt.ru>



Смотрите также дополнительный материал по DCP++ и oneAPI. Подробнее о других возможностях oneAPI [4], введение в DCP++ [5], OpenCL, SYCL and DPC++ at Intel [6].

http://www.kimrt.ru/index/course_stm/0-24

Проект реализуется победителем Конкурса на предоставление грантов преподавателям магистратуры 2020/2021 благотворительной программы «Стипендиальная программа Владимира Потанина» Благотворительного фонда Владимира Потанина.

Источники

1. Reinders, J., Ashbaugh, B., Brodman, J., Kinsner, M., Pennycook, J., & Tian, X. (2021). Data Parallel C++. Apress. <https://doi.org/10.1007/978-1-4842-5574-2>
2. Intel oneAPI Base toolkit. URL: <https://www.intel.com/content/www/us/en/developer/tools/oneapi/base-toolkit.html>
3. Get started with oneAPI. URL: <https://www.intel.com/content/www/us/en/develop/documentation/get-started-with-intel-oneapi-base-windows/top.html>
4. Intel oneAPI — один за всех, теперь — и для вас. URL: <https://habr.com/ru/company/intel/blog/476514/>
5. Введение в Data Parallel C++. Пишем первую программу. URL: <https://habr.com/ru/company/intel/blog/533126/>
6. OpenCL, SYCL and DPC++ at Intel. URL: <https://www.youtube.com/watch?v=fAVOPJLu2qA&t=3907s>