Программист C

Длительность курса: 156 академических часов

1 Современный С

1 Инструментарии разработки

Цели занятия:

познакомиться и понять как будем работать;
объяснять назначение языка С в современном мире;
перечислять современные компиляторы языка С;
рассмотреть основное назначение и различия сред разработки на языке С.

Краткое содержание:

место языка С в современном мире;
популярные компиляторы и среды разработки;
инструментарий разработки под UNIX-подобными ОС.

---

2 Основные конструкции

Цели занятия:

научиться работать с основными выражениями и операторами языка С;
научиться применять битовые операции;
научиться использовать указатели при обращении к данным.

Краткое содержание:

выражения и операторы;
битовые операции;
указатели.

---

3 Типы данных

Цели занятия:

перечислить классические типы данных;
создать собственные типы данных: структуры, объединения и перечисления;
объяснить, как данные разных типов данных хранятся в памяти.

Краткое содержание:

представление типов данных в памяти;
классические и пользовательские типы данных;
структуры, объединения, перечисления.

Домашние задания

1 Типы данных

Цель: В процессе выполнения ДЗ вы получите навык работы с бинарными файлами, описываемыми структурами языка C.

1. Написать программу, определяющую является ли заданный файл т.н. Rarjpeg-ом (изображением, в конец которого дописан архив), и выводящую список файлов в архиве, если заданный файл таковым является.
2. Для простоты подразумевается формат архива zip, а не rar.
3. Сторонние библиотеки не использовать.

Требования:
1. Создано консольное приложение, принимающее аргументом командной строки входной файл.
2. Приложение корректно определяет файл, не содержащий внутри архив (файл для проверки прилагается).
3. Приложение корректно определяет файл, содержащий внутри архив, и выводит список файлов внутри архива (файл для проверки прилагается).
4. Код компилируется без предупреждений с ключами компилятора `-Wall -Wextra -Wpedantic -std=c11`.
5. Далее успешность определяется ревью кода.

4 **Типы данных (продолжение)**

Цели занятия:

перечислить типы данных в C;
корректно работать со строками в различных кодировках;
объяснить, как данные разных типов данных хранятся в памяти.

Краткое содержание:

массивы;
строки: кодировки, Unicode;
функции;
файлы.

Домашние задания

1 Статические структуры данных

Цель: В процессе выполнения ДЗ вы получите навык работы с различными текстовыми кодировками.

1. Написать конвертор из заданной кодировки (одна из CP-1251, KOI8-R, ISO-8859-5) в UTF-8.
2. Сторонние библиотеки, включая iconv, не использовать.

Требования:
1. Создано консольное приложение, принимающее аргументами командной строки входной файл, заданную кодировку и выходной файл.
2. Конвертация из каждой из трёх указанных кодировок корректно отрабатывает (файлы для проверки прилагаются).
3. Приложение корректно обрабатывает ошибки доступа к файлам.
4. Код компилируется без предупреждений с ключами компилятора `-Wall -Wextra -Wpedantic -std=c11`.
5. Далее успешность определяется ревью кода.

5 **Стандарты C90/C99/C11**

Цели занятия:

знать о возможностях и особенностях различных версий стандарта языка C;
использовать расширения языка;
иметь представление о концепциях unspecified behaviour и undefined behaviour.

Краткое содержание:

история стандартов C;
расширения языка;
понятия unspecified и undefined behaviour.

---

6 **Современные практики программирования на C**

Цели занятия:

владеть лучшими практиками написания кода на C и декомпозиции бизнес-логики;
избегать "вредные привычки" и распространённые ошибки при кодировании на C.

Краткое содержание:

инструменты анализа и форматирования кода на C;
наиболее распространённые руководства по программированию на C;
лучшие практики, лайфхаки и соглашения оформления C-шного кода.

2 Сложные структуры данных

1 Динамические структуры данных

Цели занятия:

объяснить, как динамически распределяется память под объекты;
создать объекты: очередь, стек, связанные списки;
рассмотреть разницу между очередью и стеком;
рассмотреть разницу между односвязными и двусвязными списками;
работать с узлами списка; добавлять, удалять;

Краткое содержание:

динамическое распределение памяти;
принцип FIFO и LIFO;
очереди;
стеки;
односвязные/двусвязные списки;
деревья.

2 **Алгоритмы поиска и сортировки**

Цели занятия:

использовать различные виды сортировок;
строить алгоритмы сортировки и поиска в отсортированном массиве.

Краткое содержание:

нотация "О большое";
виды сортировок;
поиск информации в неотсортированном и отсортированном массиве.

Домашние задания

1 Структуры данных

Цель: В этом ДЗ вы получите навык создания и использования структур данных.

1. Написать реализацию хэш-таблицы с открытой адресацией со строками в качестве ключей и целыми числами в качестве значений.
2. На основе полученной реализации написать программу, подсчитывающую частоту слов в заданном файле.
3. Сторонние библиотеки не использовать.

Требования:
1. Создано консольное приложение, принимающее аргументом командной строки входной файл.
2. Приложение корректно обрабатывает ошибки доступа к файлу.
3. Приложение корректно подсчитывает и выводит на экран информацию о том, сколько раз в файле встречается каждое слово, которое есть в файле.
4. Код компилируется без предупреждений с ключами компилятора `-Wall -Wextra -Wpedantic -std=c11`.
5. Далее успешность определяется ревью кода.

3 Методики программирования

1 Библиотеки языка C

Цели занятия:

создавать собственные библиотеки;
использовать один и тот же код в различных приложениях.

Краткое содержание:

понятия статических и динамических библиотек;
компоновка программ на C;
популярные библиотеки на C с открытым исходным кодом;
соглашения вызова функций из библиотек;
понятие искажения имен функций;
инструментарий для создания, компоновки и анализа библиотек;
динамическая загрузка библиотек&

Домашние задания

1 Библиотеки языка C

Цель: В процессе выполнения ДЗ вы получите навык работы со сторонними прикладными библиотеками.

Написать программу, скачивающую с помощью [libcurl](https://curl.se/libcurl) и разбирающую с помощью произвольной сторонней библиотеки для JSON в C текущие погодные данные с [API Metaweather](https://www.metaweather.com/api/) для заданного аргументом командной строки города.

Требования:
1. Выбрана библиотека для работы с JSON в C.
2. Создано консольное приложение, принимающее аргументом командной строки название города (например, `Moscow`).
3. Приложение выводит на экран прогноз погоды на текущий день: текстовое описание погоды, направление и скорость ветра, диапазон температуры.
4. Приложение корректно обрабатывает ошибочно

заданную локацию и сетевые ошибки.
5. Код компилируется без предупреждений с ключами компилятора `-Wall -Wextra -Wpedantic -std=c11`.
6. Далее успешность определяется ревью кода.

---

2 **Препроцессор**

Цели занятия:

описать назначение макросов и способы их создания;
использовать на практике сложные макросы;
владеть директивами препроцессора.

Краткое содержание:

возможности макросов в C;
лучшие практики использования макросов;
директивы препроцессора.

---

3 **ООП в С**

Цели занятия:

научиться применять приёмы ООП в коде на С;
применять библиотеку GLib на практике.

Краткое содержание:

объектно-ориентированные приёмы в языке С;
реализации ООП в С;
фреймворки на основе библиотеки GLib.

Домашние задания

1 ООП

Цель: В процессе выполнения ДЗ вы получите навык работы с библиотеками GLib и GStreamer.

1. Написать плагин с элементом для GStreamer, воспроизводящим 16-битные little-endian несжатые моно wav-файлы.
2. Сторонние библиотеки (кроме GLib/GStreamer) не использовать.

Требования:
1. Создана динамическая библиотека, успешно подгружаемая в пайплайн GStreamer командной приблизительно следующего вида: gst-launch my-wav-element location=test.wav ! audio/x-raw,format=S16LE,channels=1,rate=48000 ! autoaudiosink.
2. Тестовый пайплайн с участием созданного модуля успешно проигрывает звуковой файл (файл для проверки прилагается).
3. Код компилируется без предупреждений с ключами компилятора -Wall -Wextra -Wpedantic -std=c11.
4. Далее успешность определяется ревью кода.

4 **Обработка ошибок**

Цели занятия:

использовать распространенные практики обнаружения и обработки ошибок в коде.

Краткое содержание:

best practices обнаружения и обработки ошибок;
механизмы assert и static_assert;
механизм setjmp/longjmp.

Домашние задания

1 Библиотека для журналирования

Цель: В процессе выполнения ДЗ вы познакомитесь с механизмами postmortem-отладки.

1. Написать библиотеку для журналирования, выводящую в лог-файл заданные сообщения с заданным уровнем важности (один из `debug`, `info`, `warning`, `error`) и с местом в коде, в котором была вызвана функция журналирования.
2. Для уровня `error` необходимо также выводить текущий стек вызовов.

Требования:
1. Создана статическая библиотека, содержащая интерфейс инициализации (для указания логфайла) и интерфейс журналирования.
2. Создано тестовое консольное приложение, демонстрирующее работу библиотеки.
3. **Бонусные баллы** за пригодность для использования при аварийных ситуациях (как-то `malloc`, вернувший `NULL`).
4. **Бонусные баллы** за потокобезопасность.
5. Код компилируется без предупреждений с ключами компилятора `-Wall -Wextra -Wpedantic -std=c11`.
6. Далее успешность определяется ревью кода.

---

5 **Основы ассемблера**

Цели занятия:

понимать, во что транслируются типичные конструкции языка C на уровне ассемблера;

рассуждать о взаимодействии программы с памятью и процессором на низком уровне;
уметь писать код на ассемблере и вставлять его в программы на C.

Краткое содержание:

архитектура центрального процессора;
модели адресации памяти;
кольца защиты;
взаимодействие процессов с памятью и процессором;
синтаксис ассемблера x86;
наиболее распространённые мнемоники ассемблера x86;
понятие ABI;
понятия кадра стека, пролога и эпилога функций.

Домашние задания

1 Ассемблерный код

Цель: Получить навык чтения ассемблерного кода.

Изучить предоставленный ассемблерный листинг, понять, над какой структурой данных в нём происходит работа и написать аналогичную программу на C, повторяющую вывод оригинальной программы.

1. Создано консольное приложение, выводящее при запуске тот же текст, что и предоставленная в листинге main.asm программа.
2. Приложение работает с той же структурой данных, что и оригинальная программа.
3. Код компилируется без предупреждений с ключами компилятора `-Wall -Wextra -Wpedantic -std=c11`.
4. **Бонусные баллы** за указание того, каким типичным функциям высшего порядка соответствуют ассемблерные процедуры `m` и `f`.
5. **Бонусные баллы** за оптимизацию процедур `m` и `f` в ассемблерном коде.
6. **Бонусные баллы** за исправление утечки памяти под структуры данных в ассемблерном коде.
7. Далее успешность определяется ревью кода.

6	**Консультация по вопросам ДЗ (Q&A)**	**Цели занятия:** получить ответы на вопросы по ДЗ **Краткое содержание:** типичные ошибки при выполнении ДЗ; преподаватели ответят на ваши вопросы.

4 Среда UNIX

1 **История, философия и основы работы в UNIX**

Цели занятия:

рассмотреть состав и основы работы в UNIX.

Краткое содержание:

обоснование, состав и философия стандарта POSIX.

2 **Стандарт POSIX и программирование под UNIX**

Цели занятия:

создавать программы для POSIX-совместимых ОС с учетом лучших практик;
использовать на практике инструментарий UNIX для разработки программ.

Краткое содержание:

инструментарий разработчика в UNIX.

Домашние задания

1 POSIX

Цель: В процессе выполнения ДЗ вы получите навык поиска утечек памяти с помощью `valgrind`.

1. Найти и исправить утечки памяти в программе из приложенного архива.
2. Для запуска `valgrind` следует использовать следующие команды в каталоге с распакованным архивом:

```shell
cd test/package
make valgrind
```

Требования:
1. Найдена как минимум одна утечка памяти, создан patch, исправляющий утечку.
2. **Бонусные баллы** за дополнительные найденные и исправленные утечки.
3. Далее успешность определяется ревью кода.

3 **Введение в процессы**

Цели занятия:

владеть механизмами обмена данными с ОС;
иметь представление о размещении данных в памяти процессов.

Краткое содержание:

взаимодействие процессов с ОС;
организация памяти процессов;
паттерны контроля за процессами.

---

4 **Демоны UNIX**

Цели занятия:

описать понятие демонов, механизмы их создания и особенности работы.

Краткое содержание:

понятие демонов;
способы создания программ-демонов;
журналирование, конфигурирование и запуск программ-демонов.

5 Процессы в C

1 Взаимодействие между процессами

Цели занятия:

создавать многопроцессные приложения с обменом данными между процессами.

Краткое содержание:

протоколы создания дочерних процессов;
сигналы UNIX;
полудуплексные каналы;
именованные каналы;
UNIX domain sockets.

Домашние задания

1 Создание демонов

Цель: В процессе выполнения ДЗ вы получите навыки создания демонов и взаимодействия через сокеты домена UNIX.

1. Написать демон получения размера заданного через конфигурацию файла.
2. Демон должен отдавать текущий размер файла по запросу через сокет домена UNIX, заданный в конфигурации, и сразу разрывать соединение с клиентом.

Требования:
1. Созданное приложение успешно запускается как в фоне, так и без демонизации.
2. Демон корректно отдаёт размер заданного файла до и после изменения файла.
3. Демон корректно обрабатывает ошибки доступа к файлу.
4. Код компилируется без предупреждений с ключами компилятора `-Wall -Wextra -Wpedantic -std=c11`.
5. Далее успешность определяется ревью кода.

2 Разделяемая память

Цели занятия:

создавать полнофункциональные многопроцессные приложения;
оперировать файлами, превышающими по размеру объём оперативной памяти.

Краткое содержание:

очереди сообщений System V;
семафоры System V и POSIX;
разделяемая память System V и POSIX;
системные вызовы mmap и msync.

Домашние задания

1 Программа, подсчитывающая контрольную сумму crc32

Цель: В процессе выполнения ДЗ вы получите навык обработки файлов, превышающих по объёму оперативную память.

1. Написать программу, подсчитывающую контрольную сумму [CRC32](https://en.wikipedia.org/wiki/Cyclic_redundancy_check#CRC-32_algorithm) (не Adler32) для файлов, превышающих по объёму оперативную память без многократного вызова `read`.
2. Сторонние библиотеки не использовать.

Требования:

1. Создано приложение, принимающее аргументами командной строки путь к файлу.
2. Приложение корректно подсчитывает CRC32 для заданного файла (файл для проверки прилагается).
3. Приложение корректно обрабатывает ошибки доступа к файлам.
4. Код компилируется без предупреждений с ключами компилятора `-Wall -Wextra -Wpedantic -std=c11`.
5. Далее успешность определяется ревью кода.

6 Многопоточность в C

1 **Введение в потоки**

Цели занятия:

рассмотреть принцип создания потоков;
объяснять различие между процессами и потоками;
понимать модели многопоточных приложений;
создавать многопоточные приложения с использованием POSIX threads.

Краткое содержание:

понятие потоков;
модели многопоточных приложений;
закон Амдала;
API библиотеки POSIX threads;
понятие примитива синхронизации.

Домашние задания

1 Многопоточный разбор логов HTTP-сервера

Цель: В процессе выполнения ДЗ вы получите опыт создания многопоточных приложений.

Написать программу, в заданное число потоков разбирающую логи веб-сервера в стандартном (["combined"] (http://httpd.apache.org/docs/2.4/logs.html#combined)) формате и подсчитывающую агрегированную статистику: общее количество отданных байт, 10 самых "тяжёлых" по трафику URL'ов и 10 наиболее часто встречающихся Referer'ов.

Требования:
1. Создано консольное приложение, принимающее аргументами командной строки директорию с логами и количество потоков.
2. Приложение корректно подсчитывает требуемые статистические данные (файлы логов для проверки прилагаются).
3. Приложение корректно обрабатывает случай пустой директории с логами.
4. Приложение корректно обрабатывает ошибки доступа к файлам.
5. Код компилируется без предупреждений с

		ключами компилятора `-Wall -Wextra -Wpedantic -std=c11`. 6. Далее успешность определяется ревью кода.
2	**Механизмы синхронизации**	**Цели занятия:** создавать многопоточные приложения с обменом данными между потоками; **Краткое содержание:** понятие состояния гонки; pthreads mutex; понятие взаимной блокировки; pthreads r/w locks; pthreads condition variables; pthreads spinlocks and barriers; механизм TLS.
3	**Функции в многопоточных приложениях**	**Цели занятия:** писать код, корректно работающий в многопоточной среде; использовать возможности стандарта C11 для создания многопоточных программ. **Краткое содержание:** понятия реентрабельности, потокобезопасности и атомарности; потоки и примитивы синхронизации в стандарте C11; атомарные типы и операции над ними в C11.

7 Сетевое взаимодействие

1 Сетевое взаимодействие

Цели занятия:

создавать приложения, взаимодействующие по сети.

Краткое содержание:

API сокетов Беркли;
уровни модели OSI в API сокетов Беркли;
механизм разрешения адресов;
обмен данными по сети;
разработка программ сервера и клиента;
Winsock;
сетевые библиотеки для языка C.

Домашние задания

1 Сетевые клиенты

Цель: В этом ДЗ вы получите опыт написания сетевых клиентов.

1. Написать программу, конвертирующую текст в ASCII art с помощью telnet-сервиса https://telehack.com (в терминале telnet telehack.com, затем отдать команду figlet).
2. Сторонние библиотеки не использовать.

Требования:
1. Создано приложение, принимающее аргументами командной строки название шрифта и текст для конвертации.
2. Приложение корректно обрабатывает сетевые ошибки доступа к сервису.
3. Код компилируется без предупреждений с ключами компилятора `-Wall -Wextra -Wpedantic -std=c11`.
4. Далее успешность определяется ревью кода.

---

2 Асинхронные сетевые интерфейсы

Цели занятия:

использовать асинхронные сетевые API в клиентских приложениях;

создавать серверные приложения, способные к работе в высоконагруженных сценариях.

Краткое содержание:

принцип мультиплексирования ввода-вывода;
системные вызовы select и poll;
POSIX asynchronous I/O;
современные асинхронные интерфейсы epoll и kqueue;
интерфейс io_uring;
Windows I/O completion ports;
библиотеки мультиплексирования ввода-вывода;
высокоуровневое асинхронное программирование, корутины.

Домашние задания

1 Асинхронный HTTP-сервер

Цель: В процессе выполнения ДЗ вы получите опыт создания асинхронных серверов.

1. Написать HTTP-сервер на основе одного из современных механизмов мультиплексирования ввода-вывода (epoll, kqueue), раздающий файлы из заданной директории.
2. Сторонние библиотеки не использовать.

Требования:
1. Создано приложение, принимающее аргументами командной строки рабочую директорию и пару адрес:порт для прослушивания.
2. Сервер корректно отдаёт файлы из заданной директории по HTTP.
3. Сервер корректно отвечает 404 статус-кодом на запросы несуществующих файлов и 403 статус-кодом на запросы файлов, на чтение которых у процесса не хватает прав.
4. **Бонусные баллы** за решение, обеспечивающее наибольшее RPS.
5. Код компилируется без предупреждений с ключами компилятора `-Wall -Wextra -Wpedantic -std=c11`.
6. Далее успешность определяется ревью кода.

3 **Веб-сервисы**

Цели занятия:

создавать веб-сервисы на C.

Краткое содержание:

основы протокола HTTP;
интерфейсы CGI и FastCGI;
веб-сервер Nginx;
основы REST API;
основы протокола Websocket;
особенности протокола HTTP/2;
библиотеки и фреймворки для создания веб-серверов на C.

8 Микроконтроллеры

1 **Введение в микроконтроллеры**

Цели занятия:

рассмотреть современные микроконтроллеры.

Краткое содержание:

современные микроконтроллеры, их архитектура; введение в Ардуино.

---

2 **Введение в Ардуино**

Цели занятия:

рассмотреть возможности Ардуино

Краткое содержание:

среда разработки Ардуино;
функции фрэймворка;
основные периферийные модули микроконтроллера.

---

3 **Разработка приложений на Ардуино**

Цели занятия:

научиться создавать прошивки в Ардуино.

Краткое содержание:

создание firmware-программы, использующей различные периферийные модули микроконтроллера.

9 Работа с базами данных

1 Современные СУБД

Цели занятия:

описать современные СУБД и тенденции их развития.

Краткое содержание:

виды СУБД: SQL, noSQL и newSQL СУБД;
транзакции;
технологии ASID и BASE: назначение и основные отличия.

---

2 Работа с СУБД из C

Цели занятия:

создавать приложения, взаимодействующие с реляционными СУБД;
создавать приложения, взаимодействующие с NoSQL-решениями.

Краткое содержание:

внутреннее устройство и встраивание SQLite в приложения;
взаимодействие с MySQL;
взаимодействие и расширение PostgreSQL;
взаимодействие с NoSQL БД: Memcached, Redis, MongoDB;
обзор NoSQL-решений на C.

Домашние задания

1 Интеграция C и СУБД

Цель: В процессе выполнения ДЗ вы получите опыт взаимодействия с базами данных.

Написать программу, вычисляющую статистические параметры в заданной таблице БД по заданной колонке:
* среднее
* максимум
* минимум

* дисперсия
* сумма

Требования:
1. Создано консольное приложение, принимающее аргументами командной строки название базы данных, название таблицы и название колонки.
2. Приложение корректно вычисляет требуемые значения по заданной колонке.
3. Приложение корректно обрабатывает нечисловой тип данных в заданной колонке.
4. Приложение корректно обрабатывает ошибки доступа к БД.
5. **Бонусные баллы** за поддержку различных движков БД.
6. Код компилируется без предупреждений с ключами компилятора `-Wall -Wextra -Wpedantic -std=c11`.
7. Далее успешность определяется ревью кода.

10 Создание игр

1 Библиотеки DirectX и OpenGL

Цели занятия:

создавать простые графические приложения на C

Краткое содержание:

обзор спецификации OpenGL;
вспомогательные библиотеки для OpenGL-приложений;
низкоуровневые примитивы, используемые для отрисовки графики;
виды проекций.

Домашние задания

1 Написать программу, отображающую вращающийся куб.

Цель: В процессе выполнения ДЗ вы получите опыт работы с графическими API.

Написать программу, отображающую вращающийся куб.

Требования:
1. Создано приложение, запускающее окно и отображающее в нём вращающийся с течением времени куб.
2. **Бонусные баллы** за наличие текстуры на кубе.
3. **Бонусные баллы** за выход из приложения по нажатию Esc.
4. Код компилируется без предупреждений с ключами компилятора `-Wall -Wextra -Wpedantic -std=c11`.
5. Далее успешность определяется ревью кода.

2 **Работа с устройствами ввода**

Цели занятия:

создавать игровые приложения, реагирующие на воздействия пользователя через устройства ввода.

Краткое содержание:

понятие игрового цикла;
обзор middleware-библиотек SDL и liballegro;
паттерны обработки событий клавиатуры, мыши и джойстика в middleware-библиотеках.

3 **Интерактивность в играх**

Цели занятия:

создавать интерактивные игровые приложения на C.

Краткое содержание:

способы задания игрового цикла;
паттерн Entity-Component-System;
обзор игровых UI фреймворков, паттерн immediate mode;
ресурсы для разработки игр на C.

Домашние задания

1 Создание игр

Цель: В процессе выполнения ДЗ вы получите опыт создания игр.

Написать несложную игру, на выбор:
* тетрис
* сапёр
* 2048
* крестики-нолики
* четыре-в-ряд
* ping-pong
* space invaders
* pacman

Требования:
1. Создано игровое приложение с главным меню и основным gameplay loop.
2. **Бонусные баллы** за графическое оформление с использованием ассетов.
3. **Бонусные баллы** за звуковое оформление.
4. **Бонусные баллы** за таблицу лидеров.
5. Код компилируется без предупреждений с ключами компилятора `-Wall -Wextra -Wpedantic -std=c11`.
6. Далее успешность определяется ревью кода.

11 Проектная работа

1 **Выбор темы и организация проектной работы**

Цели занятия:

выбрать и обсудить тему проектной работы;
спланировать работу над проектом;
ознакомиться с регламентом работы над проектом.

Краткое содержание:

правила работы над проектом и специфика проведения итоговой защиты;
требования к результату проекта и итоговой документации.

Домашние задания

1 Проектная работа

Цель: * выбрать тему проекта;
* закрепить тему в чат по дз;
* получить навыки взаимодействия с opensource-сообществом;
* написать код, который не стыдно показать в портфолио.

Написать код, соответсвующий типичным требованиям к качеству (компиляция с `-Wall -Wpedantic`, использование clang-format). Опубликовать код на Github/Gitlab. По возможности получить обратную связь от сообщества.

Для защиты используйте [шаблон презентации](https://docs.google.com/presentation/d/1ugjXX1yX400tUXxBIZaLgiwlJwpl3EzQ5fF2sp3wXPY/edit?usp=sharing).
На защиту вашего проекта дается 15 минут, более подробная информация дана на вебинаре к этому занятию.
Проект сдавать через чат с преподавателем.

Вопросы задавать руководителям курса - Кравчуку Андрею и Коробкову Виктору.

2	**Консультация по проектам и домашним заданиям**	**Цели занятия:** получить ответы на вопросы по проекту, ДЗ и по курсу. **Краткое содержание:** вопросы по улучшению и оптимизации работы над проектом; затруднения при выполнении ДЗ; вопросы по программе.
3	**Защита проектных работ**	**Цели занятия:** защитить проект и получить рекомендации экспертов. **Краткое содержание:** презентация проектов перед комиссией; вопросы и комментарии по проектам.
4	**Подведение итогов курса**	**Цели занятия:** защитить проект и получить рекомендации экспертов. **Краткое содержание:** презентация проектов перед комиссией; вопросы и комментарии по проектам.