**Итоговая работа по программированию (1ПМ+1МО)**

**Минимальные критерии**

1. (ПМ и МО) Разработать программный продукт.
   1. Игра или анимация c использованием SFML.
   2. Web-сервис с использованием httplib.
   3. Чат по локальной сети с использованием httplib.
   4. Интерфейс для взаимодействия с базой данных с использованием библиотеки sqlite3.
   5. Программа с интерфейсом на Qt
   6. И т.п.
2. (ПМ и МО) Использовать объектно-ориентированный подход и современные средства языка С++:
   1. Умные указатели
   2. Исключения
   3. C++ casts
   4. Логгер
   5. Наследование и полиморфизм
3. (ПМ и МО) Оформить код с использованием пространства имен и с разбиением на файлы.
4. (ПМ) Подготовить CMake файл для автоматизации развертывания программы.
5. (ПМ) Подготовить небольшую (пару строчек) документацию по программе. Желательно оформить в формате Markdown:
   1. Системные требования (какая ОС)
   2. Как запустить для использования
   3. Как запустить для разработки (какой компилятор использовался, среда разработки)
6. (ПМ) При разработке использовать систему контроля версий Git.

Минимальные критерии должны быть соблюдены, но вы не ограничены, если хотите сделать нечто большее. Творчество приветствуется. Только будьте готовы объяснить то, что вы сделаете.

**Как будет происходить оценивание работы?**

Учитывается:

1. Соответствие работы критериям 1-6 для ПМ и 1-3 для МО.
2. Ответы студента на возможные вопросы (см. список).
3. Оценка за лабораторные работы.

Поскольку творчество приветствуется, хорошая работа может послужить поводом добавить балл к оценке.

**Допуск к экзамену (для 1ПМ, Тарачков М.В.)\***

1. Сдано 8 л/р – допуск, максимальная оценка за экзамен 5.
2. Сдано 6 л/р – допуск, максимальная оценка за экзамен 4.
3. Сдано 4 л/р – допуск, максимальная оценка за экзамен 3.
4. Сдано <4 л/р – допуска нет.

**Учебная практика (для 1ПМ, Тарачков М.В.)\***

1. Сдано 4 задач – оценка 5.
2. Сдано 3 задачи – оценка 4.
3. Сдано 1 задача – оценка 3.
4. Сдано 0 задач – зачета нет.

\*Условия получения допуска и оценки уточняйте у своих преподавателей по практике.

\*\*Преподаватели могут поставить «автомат», я это учту, и сдавать итоговую работу не придется.

**Список вопросов по основам программирования**

1. Что такое стандарт языка?
2. Сколько места в оперативной памяти занимает тип bool и почему?
3. Как работает тернарный оператор? Привести пример.
4. Что означает создать переменную «на стеке»? Какие плюсы и минусы такого подхода?
5. Какие преимущества дает хип и какие проблемы порождает?
6. Сколько места в оперативной памяти занимают указатели unsigned int\* и char\*? Почему?
7. Что хранит в себе тип char\*\*?
8. Сколько места в оперативной памяти занимает ссылка?
9. В чем разница передачи аргумента по ссылке и по значению?
10. Как вы понимаете «абстракцию»?
11. Как вы понимаете «инкапсуляцию»?
12. В чем основное отличие класса от структуры?
13. Почему многие библиотеки на С++ используют пространство имен? Привести пример потенциальной проблемы, которая может быть без использования пространства имен.
14. Объясните, что означает \*& в следующем фрагменте кода

void update(list\*& root)

{

…

}

1. Что такое компиляция и линковка?
2. В чем отличие статической и динамической библиотек?
3. Чему равен sizeof от std::vector? Почему?
4. Чему равен sizeof от std::string? Почему?
5. Что такое константный метод класса?
6. int x = -1; unsigned int y = 0; Чему равно x+y?
7. Что будет выведено в результате выполнения программы?

int a = 0;  
int b = 1;  
if(a++ && b++)  
{  
  
}  
std::cout << a << " " << b << std::endl;

1. Чему равно выражение ~p, если p = 100 и занимает в памяти 8 бит?
2. Почему ++a эффективнее a++?
3. Чем отличается [] от at в векторе?
4. Назовите правило 3х.
5. Что делает ключевое слово default?
6. Что делает ключевое слово delete?
7. Что такое делегирующий конструктор? Пример.
8. Для чего используется идиома copy&swap?
9. Напишите сигнатуру перегрузки префиксного и постфиксного инкремента.
10. Что такое Return value optimization? Пример.
11. Что такое Copy elision? Пример.
12. Друзья (friend) в С++ – это хорошо или плохо? Приведите примеры.
13. Приведите пример кода, где появляется висячая ссылка.
14. Что такое lifetime extension у ссылок? Пример.
15. Какое свойство получает переменная, которая объявлена static?
16. Объясните основную идею наследования.
17. Что означает ключевое слово protected?
18. Почему необходимо делать деструктор базового класса виртуальным?
19. Чем помогает ключевое слово override при переопределении метода базового класса в классе-наследнике?
20. Как вы понимаете идиому RAII?
21. В чем основная идея умных указателей?
22. Почему при создании умного указателя следует использовать std::make\_unique или std::make\_shared? Пример.
23. Исключение какого типа кидает new, если не может получить требуемую память от операционной системы?
24. Что такое pull request?