МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«Гжельский государственный университет»** (ГГУ)

Колледж ГГУ

Специальность 09.02.07 Информационные системы и программирование

**Лабораторная работа №7**

**По предмету «Управление проектами»**

**На тему «Метрики Холстеда»**

ВЫПОЛНИЛ:

Студент группы ИСП-0-17

Скрябин С.И.

ПРОВЕРИЛА:

Прокуронова А.Ю.

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

п. Электроизолятор

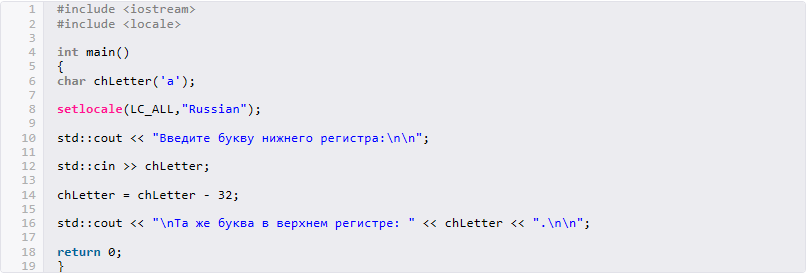
2019 г.

**Цель работы:**

Изучение основ метрической теории программ Холстеда, расчет количественных характеристик для индивидуального модуля.

**Выполнение работы:**

В пример для использования метрик Холстеда, за код возьмем небольшую программу на языке C++:



Теперь начнем анализировать и рассчитывать исходный код по метрике Холстеда:

Словарь программы - 29

Длина программы – 8

Объём программы – 30

Потенциальный (минимальный) объём – 25

Граничный объем – 35

Соотношения между операциями и операндами – 1

Уровень программы – 1.2

Интеллектуальное содержание – 53.33

Работа по программированию (общее число элементарных мысленных различий требуемых для порождения программы) – 2.46

Приближенное время программирования – 1,5 мин.

Уровень языка – 12.195

Число переданных ошибок в программе – 0

**Контрольные вопросы:**

1. Где можно использовать метрики Холстеда?

2. Чем определяются характеристики программы?

3. Как оценить качество реализации алгоритма по метрикам?

4. В чем недостаток программометрии?

**Ответы:**

1. В программировании для решения следующих задач:  
-предсказание условий, необходимых для программирования по предложенным проектам;

-определение норм первоначальных ошибок;

-количественная оценка языков программирования и эффекта модульности;

-обоснование метода измерения различий между программами, написанными специалистами

2. Словарь программы, Длина программы, Объем программы, Потенциальный объем, Граничный объем, Соотношения между операциями и операндами, Уровень программы, Интеллектуальное содержание, Работа по программированию, Приближенное время программирования, Уровень языка и Число переданных ошибок в программе.

3. Качество реализации алгоритма в большей степени определяется следующими параметрами: интеллектуальное содержание, работа по программированию, приближенное время программирования, уровень языка, уравнение ошибок.

4. Неэтичность: утверждается, что неэтично судить о производительности программиста по метрикам, введённым для оценки эффективности программного кода.

Замещение «управление людьми» на «управление цифрами», которые не учитывает опыт сотрудника и их другие качества.

Искажение: Процесс измерения может искажён за счёт того, что сотрудник знают об измеряемых показателях и стремятся оптимизировать эти показатели, а не свою работу.

Неточность: Нет метрик, которые были бы одновременно и значимы и достаточно точны.