

Краснодарское высшее военное училище имени генерала армии С.М. Штеменко
3 факультет (защиты информации на объектах информатизации)
34 кафедра (защиты информации специальными методами и средствами)

УТВЕРЖДАЮ

Врио начальника 34 кафедры (защиты информации
специальными методами и средствами)
подполковник

П.Новиков

« » августа 2025 г.

**Варианты заданий на курсовую работу по учебной дисциплине
«Языки и технологии программирования»**

Обсуждены на заседании ПМК 2 34 кафедры
протокол № 1 от «28» августа 2025 г.

Краснодар 2025

ЗАДАНИЕ
на курсовую работу по учебной дисциплине
«Языки и технологии программирования»

Тема работы: «Разработка специализированного программного обеспечения».

Цель работы:

1. Закрепление знаний о жизненном цикле специализированного программного обеспечения.
2. Получение практических навыков разработки специализированного программного обеспечения.
3. Приобретение опыта создания конструкторской и технической документации.
4. Подготовка к дипломному проектированию.

Параметры изделия должны соответствовать индивидуальному буквенно-цифровому коду задания, выданному руководителем курсовой работы.

Значение позиций буквенно-цифрового кода.

Первая цифра соответствует количеству попыток ввода некорректного пароля до реакции системы защиты на несанкционированный доступ (например, 1 – один ввод неправильного пароля, 2 – два ввода неправильного пароля и т.д.)

Первая буква обозначает метод идентификации пользователя:

A – по буквенно-цифровому паролю с возможностью его смены;

B, C, D и E по корректности функционального преобразования $f(p)$ случайно выбранного машиной числа p , если функция равна:

B – $f(p) = 2p + 3$, с возможностью смены коэффициента при p ;

C – $f(p) = p + D$, где D – текущая дата;

D – $f(p) = p - H$, где H – текущий час;

E – $f(p) = p + M$, где M – текущий месяц.

Вторая буква обозначает реакцию системы на несанкционированный доступ:

F – выдает сообщение пользователю «Неверный пароль!» и отказывает в обслуживании;

G – регистрирует число попыток ввода пароля и выход в ОС;

H – регистрирует дату и время попытки несанкционированного доступа и блокирует ПЭВМ.

Третья буква обозначает метод шифрования текстовых файлов:

J – Шифр Цезаря;

K – Шифр Виженера;

L – Шифр Атбаш;

O – Шифр Плейфера;

M – Шифр Вернама;

N – Шифр ROT1;

P – Двоичный код;

Q – метод перестановки, т.е. перестановки символов между собой по какому-либо алгоритму, за исключением вышеперечисленных.

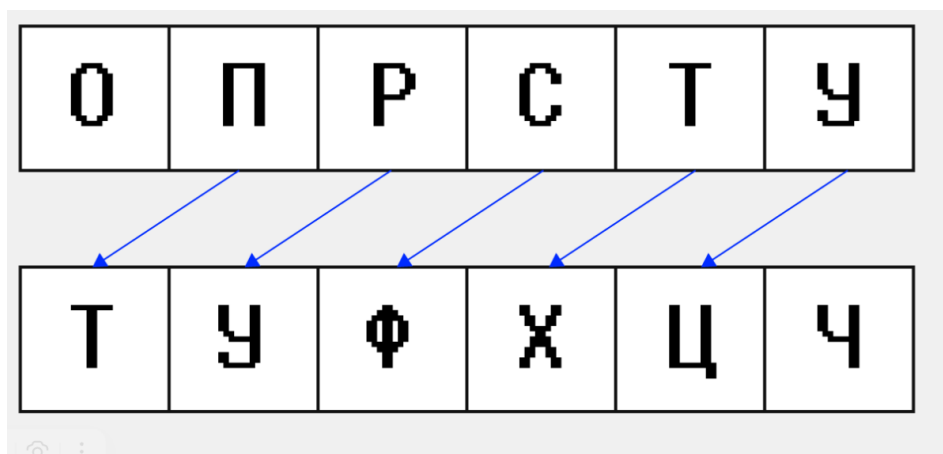
Функции зашифрования / расшифрования должны быть реализованы в динамически подключаемой библиотеке.

Методы шифрования

Шифр Цезаря: шифрует по буквам алфавита

Шифр Цезаря — это один из самых простых и старейших методов шифрования. Он был назван в честь римского императора Юлия Цезаря, который использовал его для секретной переписки.

Как работает шифр. Шифр Цезаря представляет собой моноалфавитный шифр подстановки, где каждая буква в тексте сдвигается на фиксированное количество позиций по алфавиту. Например, если сдвиг равен 3, то буква «А» становится «Г», «Б» — «Д» и так далее.



Пример шифрования. Допустим, нам нужно зашифровать слово «Привет» с использованием сдвига на три позиции:

П → переходим на три буквы вправо: П → Р → С → Т

Р: Р → С → Т → У

И: И → Й → К → Л

В: В → Г → Д → Е

Е: Е → Ж → З → И

Т: Т → У → Ф → Х

Получаем зашифрованное сообщение «тулеих». Понять его сможет только тот, кто знает, на какое количество букв в алфавите мы сдвинулись. Главный минус шифра Цезаря — его неустойчивость. Даже если человек не знает ключ, он может расшифровать сообщение методом подбора. При этом придётся перебрать не так много вариантов.

Шифр Виженера: шифрует с помощью ключа

Шифр Виженера — это полиалфавитный шифр, который развивает идеи шифра Цезаря. Его изобрёл французский дипломат Блез де Виженер в XVI веке. Он считается одним из первых методов шифрования, устойчивых к частотному анализу.

Частотный анализ — это изучение того, сколько символов встречается в том или ином массиве данных. Например, в русскоязычных текстах буква «о» встречается чаще других. Поэтому в зашифрованном тексте тоже можно найти символ, который чаще встречается, предположить, что это «о», и подобрать другие буквы. С шифром Виженера такой трюк не пройдёт.

Как работает шифр. Шифр Виженера использует ключевое слово для создания ряда шифров Цезаря, которые применяются последовательно. Каждая буква исходного текста шифруется сдвигом, соответствующим букве ключа. Если ключ короче сообщения, то он повторяется.

Пример шифрования. Зашифруем слово SKILLBOX с ключом KEY:

1. Повторяем ключевое слово KEY, чтобы длина совпадала с длиной шифруемого текста:

- Оригинальный текст: **SKILLBOX**.
- Ключевое слово: **KEYKEYKE**.

2. Теперь шифруем каждую букву, складывая позиции букв исходного текста и ключевого слова по алфавиту. После с помощью нахождения остатка от деления узнаём порядковый номер буквы в алфавите и подставляем новую букву. Помним, что всего в английском алфавите 26 букв, а для шифрования будем использовать алфавит, в котором A = 0, B = 1, ..., Z = 25.

- $S (18) + K (10) = 28 \% 26 = 2 \rightarrow C$
- $K (10) + E (4) = 14 \% 26 = 14 \rightarrow O$
- $I (8) + Y (24) = 32 \% 26 = 6 \rightarrow G$
- $L (11) + K (10) = 21 \% 26 = 21 \rightarrow V$
- $L (11) + E (4) = 15 \% 26 = 15 \rightarrow P$
- $B (1) + Y (24) = 25 \% 26 = 25 \rightarrow Z$
- $O (14) + K (10) = 24 \% 26 = 24 \rightarrow Y$
- $X (23) + E (4) = 27 \% 26 = 1 \rightarrow B$

Зашифрованное сообщение: **COGVPZYB**.

Такой шифр сложнее взломать из-за использования сразу нескольких способов обработки сообщения. Даже если попытаться подобрать ключ, то, скорее всего, ничего не выйдет.

Шифр Атбаш: шифрует в обратном порядке

Шифр Атбаш — это древний шифр замены, который использовался в еврейских текстах. Он является одним из простейших примеров моноалфавитного шифра.

Как работает шифр. Шифр Атбаш основан на инверсии алфавита: первая буква заменяется последней, вторая — предпоследней и так далее. В английском алфавите А меняется на Z, В на Y и так далее. В русском алфавите из шифра обычно исключают букву Ё.

Пример шифрования. Зашифруем слово «привет»:

- П → Р
- Р → П
- И → Ч
- В → Э
- Е → Ъ
- Т → Н

Зашифрованное сообщение: **РПЧЭЪН**.

Шифр Плейфера: шифрует с помощью матрицы

Шифр Плейфера — это один из первых примеров биграммного шифра (шифрования парами букв), который был разработан в 1854 году Чарльзом Уитстоном, но известен под именем его популяризатора, барона Плейфера.

Как работает шифр. Шифр использует матрицу 5×5 , заполненную ключевым словом и остальными буквами алфавита (буквы I и J объединяются). Сообщение делится на пары, и каждая пара шифруется в зависимости от положения букв в матрице.

Пример шифрования. Допустим, у нас есть ключевое слово KEY и мы хотим зашифровать сообщение SKILLBOX. Вписываем ключевое слово в матрицу и дополняем его оставшимися буквами алфавита так, чтобы не было повторов. Матрица будет выглядеть так:

K	E	Y	A	B
C	D	F	G	H
I/J	L	M	N	O
P	Q	R	S	T
U	V	W	X	Z

Слово, которое хотим зашифровать, делим на биграммы: SKILLBOX = SK, IL, LB, OX. После этого заменяем исходные биграммы на новые, следуя правилам:

- Если два символа биграммы **совпадают**: после первого символа добавляем букву X, считаем бигramму новой и продолжаем расшифровывать сообщение.

- Если символы биграммы встречаются **в одной строке**: заменяем на символы на ближайшие, расположенные справа.

- Если символы биграммы встречаются **в одном столбце**: заменяем на символы, которые находятся под ними в этом же столбце.

- Если символы биграммы находятся **в разных строках и столбцах**: заменяем на символы в тех же строках, но в других углах прямоугольника.

Например:

- SK → PA (разные строки и столбцы);

- IL → LM (одна строка);

- LB → OE (разные строки и столбцы);

- OX → NZ (разные строки и столбцы).

Зашифрованное сообщение: **PALMOENZ**.

Шифр Вернама: шифрует с помощью чисел

Шифр Вернама — это один из немногих шифров, который является абсолютно стойким к дешифровке. Этот шифр был разработан Гилбертом Вернамом в 1917 году.

Как работает шифр. Шифр Вернама использует одноразовый ключ, который равен по длине сообщению. Для шифрования используется операция побитового исключающего ИЛИ (XOR) между битами сообщения и ключа. При условии, что ключ используется только один раз и генерируется случайно, шифр Вернама является неуязвимым.

Пример шифрования

Возьмём слово «ПРИВЕТ» и представим его в двоичном коде, используя кодировку UTF-8 (важно помнить, что двоичное представление букв алфавита зависит от регистра):

- П = 11010000 10011111
- Р = 11010000 10100000
- И = 11010000 10011000
- В = 11010000 10010010
- Е = 11010000 10010101
- Т = 11010000 10100010

Для шифрования требуется случайный ключ той же длины в двоичном виде. Для примера в качестве ключа возьмём комбинацию 01101010 10110101. Затем применим операцию **XOR** к каждому символу слова «ПРИВЕТ». Можно воспользоваться [калькулятором](#), чтобы ускорить вычисления.

Результат XOR для первой пары:

11010000 10011111 XOR 01101010 10110101 = 10111010 00101010

Технические детали

Одноразовые ключи, равные по длине самому сообщению, делают шифр Вернама одновременно стойким к дешифрованию и не очень практичным. Если сообщение длинное, то понадобится такой же длинный ключ. Кроме того, для каждого сообщения всегда нужен новый ключ. Это создаёт проблему обмена ключами: как двум людям обменяться ключом так, чтобы его никто, кроме них, не узнал?

Шифр ROT1

Ключ прост: каждая буква заменяется на следующую за ней в алфавите. Так, А заменяется на Б, Б — на В, и т. д. Фраза «Уйрйшоьк Рспдсбннйту» — это «Типичный Программист».

Двоичный код

Текстовые данные вполне можно хранить и передавать в двоичном коде. В этом случае по таблице символов (чаще всего ASCII) каждое простое число из предыдущего шага сопоставляется с буквой: 01100001 = 97 = «а», 01100010 = 98 = «b», etc. При этом важно соблюдение регистра.

Стадии разработки программного обеспечения

Основными и неизменными стадиями и этапами разработки программного обеспечения являются техническое задание, эскизный проект, технический и рабочий проекты.

Эскизный проект. На этой стадии детально разрабатываются структуры входных и выходных данных, определяется форма их представления. Разрабатывается общее описание алгоритма, сам алгоритм, структура программы. Разрабатываются план мероприятий по разработке и внедрению программы.

Технический проект. Содержит разработанный алгоритм решения задачи, а также методы контроля исходной информации. Здесь же разрабатываются средства обработки ошибок и выдачи диагностических сообщений, определяются формы представления исходных данных и конфигурация технических средств.

Рабочий проект. На этой стадии осуществляется программирование и отладка программы, разработка программных документов, программы и методики испытаний. Подготавливаются контрольно-отладочные примеры. Окончательно оформляются документация и графический материал. Указывается, что в ходе разработки программы должна быть подготовлена следующая документация:

- описание программы;**
- текст программы;**
- программа и методика испытаний;**
- руководство оператора.**

Оформление отчетных материалов

Результаты курсовой работы оформляются в виде готового программного комплекса в соответствии с требованиями ГОСТ и ЕСПД и должны содержать следующие документы и соответствовать требованиям изложенным **в приложении 2** к настоящему заданию.

Описание программы (ГОСТ 19.402-78)

Описание программы обязательно должно включать информационную часть – аннотацию и содержание.

Основная часть документа должна состоять из следующих разделов:

- общие сведения;
- функциональное назначение;
- описание логической структуры;
- используемые технические средства;
- вызов и загрузка;
- входные данные;
- выходные данные.

В зависимости от особенностей программы допускается вводить дополнительные разделы или объединять отдельные разделы.

В разделе «Общие сведения» должны быть указаны:

- обозначение и наименование программы;
- программное обеспечение, необходимое для функционирования программы;
- языки программирования, на которых написана программа.

В разделе «Функциональное назначение» должны быть указаны классы решаемых задач и (или) назначение программы и сведения о функциональных ограничениях на применение. Например: «Программа предназначена для решения задач ... Программа представляет собой ядро автоматизированного рабочего места ... Пользователь имеет возможность ..., осуществить ..., запустить ..., проанализировать ..., получить результаты анализа и обработки ..., построить ... и т.п.».

В разделе «Описание логической структуры» должны быть указаны:

алгоритм программы (оформляется в соответствии с требованиями **ГОСТ 19.701-90**. Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения);

используемые методы;

структура программы с описанием функций составных частей и связи между ними;

связи программы с другими программами.

Описание логической структуры программы выполняют с учетом текста программы на исходном языке.

В разделе «Используемые технические средства» должны быть указаны типы электронно-вычислительных машин и устройств, которые используются при работе программы.

В разделе «Вызов и загрузка» должны быть указаны:

способ вызова программы с соответствующего носителя данных;

входные точки в программу.

Допускается указывать адреса загрузки, сведения об использовании оперативной памяти, объем программы.

В разделе «Входные данные» должны быть указаны:

характер, организация и предварительная подготовка входных данных;

формат, описание и способ кодирования входных данных.

В разделе «Выходные данные» должны быть указаны:

характер и организация выходных данных;

формат, описание и способ кодирования выходных данных.

Допускается содержание разделов иллюстрировать пояснительными примерами, таблицами, схемами, графиками.

В приложение к описанию программы допускается включать различные материалы, которые нецелесообразно включать в разделы описания.

Текст программы (ГОСТ 19.401-78)

Требования к оформлению текста программы достаточно просты и естественны для грамотного программиста. Основное, чем требуется руководствоваться при создании этого документа – это то, что текст программы должен быть удобочитаемым.

Составление информационной части (аннотация и содержание) является необязательным.

Основная часть документа должна состоять из текстов одного или нескольких разделов, которым даны наименования.

Допускается вводить наименования также и для совокупности разделов.

Каждый из этих разделов реализуется одним из типов символической записи,

например:

- символическая запись на исходном языке;
- символическая запись на промежуточных языках;
- символическое представление машинных кодов и т.п.

В символическую запись разделов рекомендуется включать комментарии, которые могут отражать, например, функциональное назначение, структуру.

Текст каждого программного файла начинается с «шапки», в которой указывается: наименование программы, автор, дата создания программы, номер версии, дата последней модификации.

Обязательными являются комментарии, а также строгое соблюдение правил отступа.

В листинге 1 приведен пример подобного хорошо читаемого текста программы.

```
/* Исходные тексты Windows'98 Автор: Nobody Really Source Code to Windows 98 */
```

```
#include "win31.h"
#include "win95.h"
#include "evenmore.h"
#include "oldstuff.h"
#include "billrulz.h"
#include "monopoly.h"
#define INSTALL = HARD
char make_prog_look_big[1600000];
void main()
{
    while(!CRASHED)
    {
        display_copyright_message();
        display_bill_rules_message();
        do_nothing_loop();
        if(first_time_installation)
        {
            make_50_megabyte_swapfile();
            do_nothing_loop();
            totally_screw_up_HPFS_file_system();
            search_and_destroy_the_rest_of_OS/2();
            disable_Netscape();
            disable_RealPlayer();
            disable_Corel_Products();
            hang_system();
        }
        write_something(anything);
        display_copyright_message();
        do_nothing_loop();
        do_some_stuff();
        if (still_not_crashed)
        {
            display_copyright_message();
            do_nothing_loop();
            basically_run_windows_3.1();
            do_nothing_loop();
            do_nothing_loop();
        }
    }
}
```

```

    }
}
if (detect_cache())
    disable_cache();
if (fast_cpu())
{
    set_wait_states(lots);
    set_mouse(speed, very_slow);
    set_mouse(action, jumpy);
    set_mouse(reaction, sometimes);
}
/* printf("Welcome to Windows 3.11"); */
/* printf("Welcome to Windows 95"); */
printf("Welcome to Windows 98");
if (system_ok())
    crash(to_dos_prompt)
else
    system_memory = open("a:\swp0001.swp", O_CREATE);
while(something)
{
    sleep(5);
    get_user_input();
    sleep(5);
    act_on_user_input();
    sleep(5);
}
create_general_protection_fault();
}

```

Листинг 1

Программа и методика испытаний (ГОСТ 19.301-79)

В этом документе содержится описание того, что и как необходимо сделать, дабы убедиться в правильности работы программы. Фактически, этот документ является определяющим для приемо-сдаточных испытаний. Грамотно составленная программа и методика испытаний – это залог подписания акта сдачи-приемки, т.е. того, во имя чего вы потратили столько сил и времени.

Составление информационной части (аннотации и содержания) является необязательным.

Формально этот ГОСТ используется для разработки документов планирования и проведения испытательных работ по оценке готовности и качества программной системы.

Документ «Программа и методика испытаний» должен содержать следующие разделы:

- объект испытаний;
- цель испытаний;
- требования к программе;
- требования к программной документации;
- средства и порядок испытаний;
- методы испытаний.

В разделе «Объект испытаний» указывают наименование, область применения и обозначение испытываемой программы. Пример: «Объектом испытаний является

программа ..., предназначенная для ...».

В разделе «Цель испытаний» должна быть указана цель проведения испытаний. Пример: «Проверка надежности функционирования программы».

В разделе «Требования к программе» должны быть указаны требования, подлежащие проверке во время испытаний и заданные в техническом задании на программу. Пример: «Функционирование программы не должно приводить к сбою (фатальному нарушению работы системы). Организация диалога должна предусматривать защиту от ввода некорректных данных. Программа должна выдавать диагностику состояния системы и сообщения о любых возникших ошибках ... и т.п.».

В разделе «Требования к программной документации» должны быть указаны состав программной документации, предъявляемой на испытания, а также специальные требования, если они заданы в техническом задании на программу. Пример: «Состав программной документации, предъявляемой на испытания: описание программы (ГОСТ 19.402-78); программа и методика испытаний (ГОСТ 19.301-79); текст программы (ГОСТ 19.401-78)».

В разделе «Средства и порядок испытаний» должны быть указаны технические и программные средства, используемые во время испытаний, а также порядок проведения испытаний. Пример:

«Программа работает в соответствии с условиями эксплуатации ОС *MS Windows* (версия не ниже 7.0) на ПК типа *IBM PC/AT*, а также на совместимых с ним. Для работы необходим также адаптер *EGA (VGA)*.

Порядок проведения испытаний:

1. Запуск программы осуществляется
2. Выбирается ...
3. Нажимается ...
4. Последовательно выбираются ...»

В разделе «Методы испытаний» должны быть приведены описания используемых методов испытаний. Методы испытаний рекомендуется по отдельным показателям располагать в последовательности, в которой эти показатели расположены в разделах «Требования к программе» и «Требования к программной документации».

В методах испытаний должны быть приведены описания проверок с указанием результатов проведения испытаний (перечней тестовых примеров, контрольных распечаток тестовых примеров и т. п.).

Тестовые примеры.

Пример: «Для проведения испытаний предлагаются ..., описание которых содержатся в файлах ...Содержимое тестовых файлов и результаты работы программы приведены в Приложении 1».

Руководство оператора (ГОСТ 19.505-79)

Структуру и оформление документа устанавливают в соответствии с **ГОСТ 19.105-78**.

Составление информационной части (аннотации и содержания) является обязательным.

Руководство оператора должно содержать следующие разделы:

назначение программы;
 условия выполнения программы;
 выполнение программы;
 сообщения оператору.

В зависимости от особенностей документы допускается объединять отдельные разделы или вводить новые.

В разделе «Назначение программы» должны быть указаны сведения о назначении программы и информация, достаточная для понимания функций программы и ее эксплуатации.

В разделе «Условия выполнения программы» должны быть указаны условия, необходимые для выполнения программы (минимальный и (или) максимальный состав аппаратных и программных средств и т.п.).

В разделе «Выполнение программы» должна быть указана последовательность действий оператора, обеспечивающих загрузку, запуск, выполнение и завершение программы, приведено описание функций, формата и возможных вариантов команд, с помощью которых оператор осуществляет загрузки и управляет выполнением программы, а также ответы программы на эти команды.

В разделе «Сообщения оператору» должны быть приведены тексты сообщений, выдаваемых в ходе выполнения программы, описание их содержания и соответствующие действия оператора (действия оператора в случае сбоя, возможности повторного запуска программы и т.п.).

Допускается содержание разделов иллюстрировать поясняющими примерами, таблицами, схемами, графиками.

В приложения к руководству оператора допускается включать различные материалы, которые нецелесообразно включать в разделы руководства.

Оформление курсовой работы выполняется в строгом соответствии с требованиями к программным документам, выполненным печатным способом (ГОСТ 19.106-78) и с требованиями к программным документам (ГОСТ 19.105-78).

Титульный лист приведен в приложении 1.

При выполнении курсовой работы рекомендуется использовать среду быстрой разработки *Embarcadero RAD Studio C++ Builder*.

Все материалы курсовой работы, включая файлы проекта *Embarcadero RAD Studio C++ Builder*, представляются в электронном виде.

Презентация к докладу

Презентация, подготовленная курсантом для защиты курсовой работы, должна включать следующие слайды:

первый слайд – эмблема, наименование Училища, факультета и кафедры, гриф ограничения доступа (при необходимости), наименование работы, тема, автор, руководитель, город, год защиты КР;

второй слайд – вариант КР, со значениями позиций буквенно-цифрового кода, цель(и) работы.

третий и последующие слайды – описание программы, функциональное назначение, разработанные алгоритмы, используемые методы, структура программы и т.д.;

предпоследний слайд - видеоролик, демонстрирующий работу, разработанного СПО;

последний слайд – результаты дипломной работы.

Всего не более 10 слайдов.

Правила оформления презентации:

презентация предполагает сочетание информации различных типов: текста, графических изображений (таблицы, алгоритмы и т.д.). Необходимо учитывать специфику комбинирования фрагментов информации различных типов;

все объекты слайда (текст, рисунки, таблицы, диаграммы) должны появляться одновременно или автоматически в заданной последовательности, а смена слайдов должна происходить по щелчку манипулятора типа «мышь»;

общее требование к расположению материалов на слайде – возможность его прочесть из любого места в аудитории, удобное для глаз сочетание цветов;

важным является требование читаемости (различимости букв, цифр, линий – как по размеру, так и цвету);

текстовый фрагмент должен быть структурирован: иметь заголовок, в случае списка – каждый его элемент должен быть четко обозначен. При необходимости, ключевые слова могут быть выделены полужирным шрифтом или цветом.

В оформлении презентаций выделяют два блока: оформление слайдов и представление информации на них. Для создания качественной презентации необходимо соблюдать требования, предъявляемые к оформлению данных блоков.

Оформление слайдов презентации

Дизайн (для Microsoft Office 2007-2013)	тема: Office используется на всех слайдах; цвета: Office 2007-2013; шрифт: Arial; стиль фона: стиль 1; размер слайда: альбомный; формат фона: сплошная заливка.
Цвет фона	необходимо выбирать светлые цвета фона
Использование цвета	на одном слайде рекомендуется использовать не более трех цветов; для текста использовать контрастные цвета.
Анимация, звук	не допускается использование анимации, гиперссылок и звуковых эффектов.
Нумерация слайдов	все слайды, кроме титульного должны иметь нумерацию, шрифт номера слайда, должен быть размером № 24 пункта, полужирный; номер слайда следует располагать в правом верхнем углу слайда.

Представление информации

Содержание информации	на слайде используют короткие слова и предложения; каждый слайд, кроме титульного, должен иметь заголовок.
-----------------------	---------------------------------------------------------------------------------------------------------------

Расположение информации на странице	используется горизонтальное расположение информации; наиболее важная информация должна располагаться в центре слайда.
Текстовая информация	шрифт размером № 16-32 для заголовка, размером № 16-28 для основного текста; цвет шрифта и цвет фона должны контрастировать (текст должен хорошо читаться); тип шрифта: для заголовка и основного текста Arial; количество слов в слайде не должно превышать 30; курсив, подчеркнутый, полужирный шрифт, прописные буквы рекомендуется использовать только для смыслового выделения фрагмента текста.
Способы выделения информации	Следует использовать: рамки; границы, заливку; штриховку, стрелки; рисунки, диаграммы, схемы для иллюстрации наиболее важных фактов.
Оформление формул	рекомендуется размещать общую форму записи и/или результат, отображать всю цепочку решения не следует. Большое количество формул на слайде «не читается». На слайд выносятся только самые главные формулы, графики, величины, значения.
Графическая информация	для наглядного отображения и сравнения данных рекомендуется использовать диаграммы; при необходимости показать несколько состояний диаграммы или экранной формы предпочтительно создавать последовательность из нескольких слайдов; все рисунки, диаграммы, графики должны иметь заголовки и пояснительный текст; рисунки и диаграммы должны соответствовать содержанию; подписи к схемам, графикам и диаграммам оформляются только черным цветом шрифта; схемы, диаграммы и графики следует оформлять черными линиями толщиной от 0,25 до 1 пункта.
Оформление таблиц	в таблицах не должно быть более 5 строк и 5 столбцов – в противном случае данные в таблице будет просто невозможно увидеть, т.е. таблица будет «не читаема»; заголовки граф таблицы должна иметь отличие (например, размер шрифта) от основных данных.
Объем информации	не стоит заполнять один слайд слишком большим объемом информации.

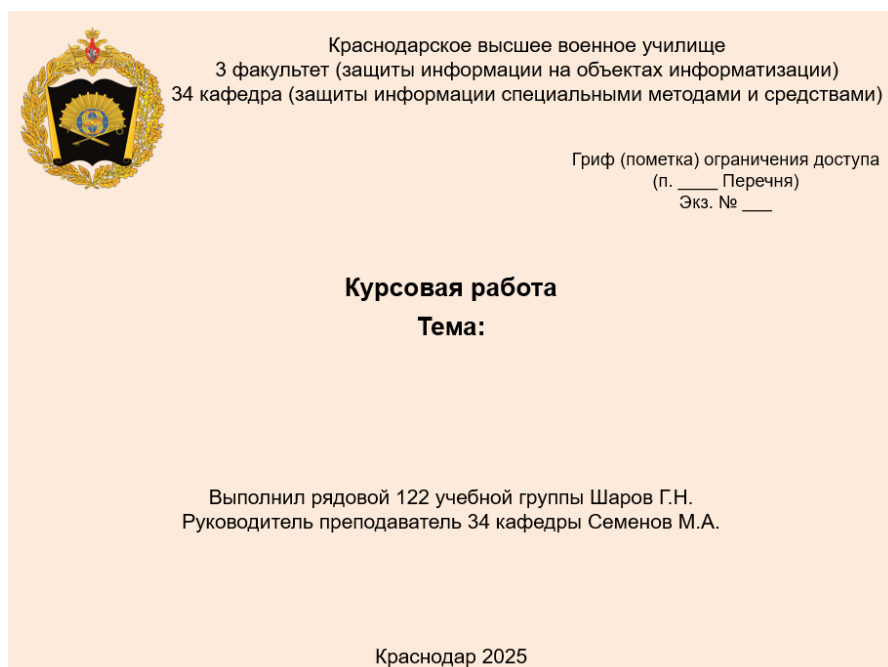
Требования к оформлению заголовков	заголовок должен находиться сверху слайда; точка в конце заголовка не ставится; не следует писать длинные заголовки; слайды не должны иметь одинаковые заголовки. Но если требуется назвать одинаково – нужно писать в конце (1), (2), (3) или Продолжение 1, Продолжение 2.
Виды слайдов	Следует использовать разные виды слайдов: с текстом; с таблицами; с графиками; с диаграммами.

Каждый новый слайд презентации должен логически вытекать из предыдущего и одновременно подготавливать появление следующего.

Слайды, сопровождающие доклад, должны обеспечить полное и точное представление материала в максимально удобной для восприятия форме в пределах отведенного времени. Они должны отражать ключевые моменты доклада, а также содержать материал, который либо трудно, либо долго описывать словами.

Курсант при изложении материала доклада должен строго следовать последовательности слайдов.

Требования к оформлению слайдов презентации



Слайд № 1 – титульный слайд.

Содержание слайда: большая эмблема (отношение высоты к ширине 1,4); наименования Училища, факультета и кафедры – размер шрифта № 16; гриф ограничения доступа (при необходимости), номер пункта Перечня, номер

экземпляра – размер шрифта № 14; наименование работы, тема – размер шрифта № 20, полужирный; автор, руководитель, город, год – размер шрифта № 16.

Вариант, цель курсовой работы
2

Вариант

Цель работы

Слайд № 2 - вариант, цель(и) работы.



Слайды № 3 - 9 – содержат таблицы, алгоритмы, видео.

РЕЗУЛЬТАТЫ РАБОТЫ

1. Проведен анализ требований ...
2. Определены содержание и последовательность работы ...
3. Предложены и обоснованы принципы, методы ...
4. Приведены варианты оформления ...
5. Разработаны рекомендации ...

Слайд № 10 – результаты работы.

Защита курсовой работы

Защита курсовой работы производится перед комиссией, состоящей из лектора потока и преподавателя, ведущего другие виды занятий.

Курсант допускается к защите при условии наличия оформленной пояснительной записки, разработанного специализированного программного обеспечения, презентации к докладу.

Для защиты курсанту отводится 10-15 минут на изложение содержания работы и демонстрации разработанного специализированного программного обеспечения, в процессе защиты комиссия высказывает свои замечания.

По результатам защиты (доклад, ответы на вопросы, качество работы) выставляется оценка в ведомости и на титульном листе пояснительной записки. В случае выявления принципиальных ошибок курсовая работа возвращается на доработку.

В случае неудовлетворительной оценки назначается повторная защита с устранением всех ошибок курсовой работы или с выдачей нового задания. При отсутствии достаточного материала по курсовой работе в контрольные сроки, курсанту, как правило, выдается новое задание.

Список рекомендованной литературы

1. Капустин, С.А. Технологии и методы программирования / С.А. Капустин, В.М. Москалев, С.Л. Шаповалов, А.Ш. Альбеков // учеб. пособие. – Краснодар: КВВУ, 2018. – 322 с.

2. Технология и методы программирования [Электронный ресурс]: учебное пособие / А.Н. Стадник, И.Д. Королев, В.В. Лозовский, С.А. Коноваленко, А.П. Нечепуренко, С.В. Иванов. – Краснодар: КВВУ, 2020. (51,3 Мб). – (34 каф.). –

(Электронная библиотека КВВУ. – Читальный зал).

3. Языки программирования [Электронный ресурс]: учебное пособие. / А.Н. Стадник, И.Д. Королев, В.В. Лозовский, С.А. Коноваленко, А.П. Нечепуренко, С.В. Иванов. – Краснодар: КВВУ, 2020. – (69 Мб). – (34 каф.) – (Электронная библиотека КВВУ. – Читальный зал).

4. Капустин, С.А. Языки программирования / С.А. Капустин, А.Ш. Альбеков, В.М. Москалев, С.А. Диченко // учеб. пособие. – Краснодар: КВВУ, 2018. – 577 с.

5. Капустин С.А., Заметаев Н.В., Альбеков А.Ш. Основы алгоритмизации и программирования: учеб. пособие. – Краснодар: КВВУ, 2016. – 289 с.

6. Шилдт, Герберт. Полный справочник по C [Текст]: [пер. с англ.] / Герберт Шилдт – 4- изд. – М.: Издательский дом «Вильямс», 2002. – 704 с.: ил. – Парал. тир. англ. – ISBN 5-84590226-6 (рус.).

7. Арнуш, Крейг. Borland C++ 5 [Текст]: Освой самостоятельно: [пер. с англ.] / Крейг Арнуш. – М.: Восточная Книжная Компания, 1997. – 720 с.: ил. – ISBN 5-89350-077-8 (в пер.).

8. Архангельский, А.Я. Программирование в C++ *Builder 5* [Текст] / А.Я. Архангельский. – М.: ЗАО «Издательство БИНОМ», 2001. – 1152 с.: ил. – ISBN 5-7989-0191-2 (рус.).

9. Седжвик, Роберт Фундаментальные алгоритмы на C++ [Текст]: Анализ / Структуры данных / Сортировка / Поиск: [пер. с англ.] / Роберт Седжвик. – К.: Издательство «ДиаСофт», 2001. – 688 с. – ISBN 966-7393-89-5 (рус.)

10. ГОСТ 19.001-77 ЕСПД. Общие положения.

11. ГОСТ 19.101-77 ЕСПД. Виды программ и программных документов (переиздан в ноябре 1987 г. с изм.).

12. ГОСТ 19.102-77 ЕСПД. Стадии разработки.

13. ГОСТ 19.103-77 ЕСПД. Обозначение программ и программных документов.

14. ГОСТ 19.104-78 ЕСПД. Основные надписи.

15. ГОСТ 19.105-78 ЕСПД. Общие требования к программным документам.

16. ГОСТ 19.106-78 ЕСПД. Требования к программным документам, выполненным печатным способом.

17. ГОСТ 19.201-78 ЕСПД. Техническое задание. Требования к содержанию и оформлению.

18. ГОСТ 19.202-78 ЕСПД. Спецификация. Требования к содержанию и оформлению.

19. ГОСТ 19.301-79 ЕСПД. Программа и методика испытаний.

20. ГОСТ 19.401-78 ЕСПД. Текст программы. Требования к содержанию и оформлению.

21. ГОСТ 19.402-78 ЕСПД. Описание программы.

22. ГОСТ 19.404-79 ЕСПД. Пояснительная записка. Требования к содержанию и оформлению.

23. ГОСТ 19.501-78 ЕСПД. Формуляр. Требования к содержанию и оформлению.

24. ГОСТ 19.502-78 ЕСПД. Описание применения. Требования к содержанию и оформлению.

25. ГОСТ 19.503-79 ЕСПД. Руководство системного программиста.

Требования к содержанию и оформлению.

26. ГОСТ 19.504-79 ЕСПД. Руководство программиста.

27. ГОСТ 19.505-79 ЕСПД. Руководство оператора.

28. ГОСТ 19.506-79 ЕСПД. Описание языка.

29. ГОСТ 19.508-79 ЕСПД. Руководство по техническому обслуживанию.

Требования к содержанию и оформлению.

30. ГОСТ 19.604-78 ЕСПД. Правила внесения изменений в программные документы, выполняемые печатным способом.

31. ГОСТ 19.701-90 ЕСПД. Схемы алгоритмов, программ, данных и систем.

Условные обозначения и правила выполнения.

32. ГОСТ 19.781-90. Обеспечение систем обработки информации программное.

Варианты заданий

220 учебная группа

№ п/п	Должность	Фамилия Имя Отчество	Вариант
1.	матрос	Абалакин Николай Николаевич	2EGQ
2.	матрос	Акимов Александр Владимирович	2CHP
3.	матрос	Андрейчук Вячеслав Евгеньевич	1CHJ
4.	матрос	Бондарев Данил Васильевич	1DGL
5.	матрос	Денисов Евгений Александрович	2EGL
6.	матрос	Иванов Максим Дмитриевич	3AFQ
7.	матрос	Иващук Илья Евгеньевич	3DGN
8.	матрос	Ищенко Родион Олегович	3DGL
9.	матрос	Калмыков Герман Андреевич	1EFL
10.	матрос	Капустин Даниил Александрович	1BHQ
11.	матрос	Константинов Сергей Андреевич	1ENO
12.	матрос	Ленченко Александр Сергеевич	1EGJ
13.	ефрейтор	Морозов Андрей Павлович	2DFM
14.	матрос	Наумик Станислав Андреевич	2AFP
15.	матрос	Новинский Владислав Дмитриевич	2EHJ
16.	матрос	Носенко Георгий Сергеевич	1ENO
17.	матрос	Обловатнев Илья Валентинович	1DGL
18.	матрос	Отрок Захар Сергеевич	2DHL
19.	матрос	Переверзев Денис Витальевич	1EFO
20.	матрос	Переходов Алексей Денисович	3AFQ
21.	ефрейтор	Редечкин Евгений Андреевич	2CHQ
22.	матрос	Редькин Матвей Геннадьевич	3CGL
23.	матрос	Рогожин Алексей Сергеевич	1AFO
24.	матрос	Саламатин Андрей Алексеевич	3AFQ
25.	матрос	Сидоренко Никита Андреевич	1BGP
26.	матрос	Тимофеев Василий Эдуардович	3CFM
27.	матрос	Тимофеев Ярослав Юрьевич	1AFO
28.	матрос	Толмачев Илья Романович	2EGO
29.	матрос	Тубольцев Никита Юрьевич	2CGO
30.	матрос	Чернов Андрей Владимирович	2EGN
31.	матрос	Шептухин Анатолий Григорьевич	1DGP

Старший преподаватель 34 кафедры (защиты информации специальными методами и средствами)

И.Ельников

Краснодарское высшее военное училище имени генерала армии С.М. Штеменко
3 факультет (защиты информации на объектах информатизации)
34 кафедра (защиты информации специальными методами и средствами)

К ЗАЩИТЕ ДОПУСКАЮ
Руководитель курсовой работы

И.Ельников

« » _____ 2025 г.

КУРСОВАЯ РАБОТА

по учебной дисциплине
«Языки и технологии программирования»

Тема работы:
«Разработка специализированного программного обеспечения»

Исполнитель: курсант 121 учебной группы
Иванов Иван Иванович

Краснодар 2025

Приложение 2
к заданию на курсовую работу
по учебной дисциплине
«Языки и технологии программирования»

Требования к разрабатываемой программе:

1. Программа должна состоять из модулей идентификации, шифрования и расшифрования при этом два последних модуля должны быть реализованы как DLL-библиотеки, количество модулей при необходимости может быть увеличено;
2. Логин пользователя формируется следующим образом – фамилия исполнителя (в английской раскладке), нижнее подчеркивание, номер по списку в строевой записке группы. Пример: *Ivanov_10*;
3. Список логинов за учебную группу оформляется в виде текстового документа в любом редакторе и хранится у заместителя командира учебной группы, при защите документ загружается в ЭВМ (рабочий стол) на которой будет происходить защита курсовой работы;
4. Проверка соответствия логина и пароля для входа в программу должен осуществляться следующим образом. Пользователь вводит в поля ввода пароля и логина соответствующие данные, затем программой производится проверка введенного пароля с паролем, который формируется программой в соответствии с вариантом на курсовую работу, логин сверяется с текстовым документом в котором перечислены все логины учебной группы.
5. В случае неверного ввода логина или пароля программа должна очистить поля и выдать сообщения соответствующее варианту задания на курсовую работу.
6. Исходный текст для шифрования (расшифрования) должен загружаться как отдельный файл, расшифрованный (зашифрованный) текст должен сохраняться в виде файла в определённой пользователем папке.

Требования к графическому интерфейсу разрабатываемой программе:

Программа должна содержать следующие окна:

1. Окно идентификации пользователя.
2. Окно смены пароля (коэффициента при Р) – при необходимости.
3. Окно для шифрования и расшифрования.

Требования по визуальному оформлению окна идентификации пользователя:

Разместить и оформить «Окно идентификации пользователя» согласно рис.1., должно быть статичным, размеры окна 700x500, исключаяющим возможность изменения его размеров.

Окно должно содержать следующие компоненты:

1. Поле ввода логина пользователя.

2. Поле ввода пароля.
3. Компонент для отражения текущей даты.
4. Компонент для отражения случайного числа.
5. Кнопка входа в программу.
6. Кнопка смены пароля (коэффициента при P) – при необходимости.

The screenshot shows a window titled "окно идентификации" (identification window). It contains four labels: "Время" (Time) at the top left, "Рандомное число" (Random number) at the top right, "Логин" (Login) on the left, and "Пароль" (Password) on the left. There are two empty text input fields on the right, one for the login and one for the password. At the bottom, there are two buttons: "ВХОД" (Login) on the left and "смена пароля" (Change password) on the right.

Рис.1 - Окно идентификации пользователя

7. При вводе неверного пароля, должно появиться всплывающее окно, в котором должна отразиться информация об оставшихся попытках ввода пароля и логина (рис. 2).

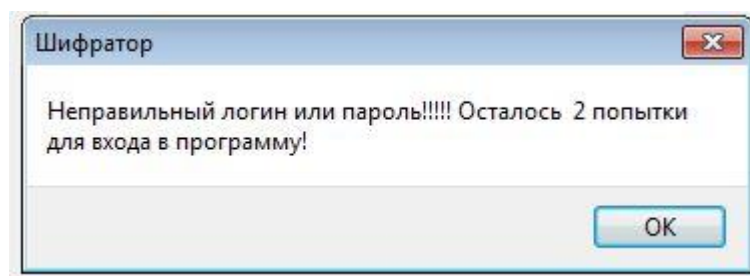


Рис.2 - Всплывающее сообщение.

8. При превышении количества неправильных ввода пароля или логина

обеспечить реагирование программы в соответствии с вариантом задания (вторая буква), либо всплывающее окно с информационным сообщением, либо блокировка системы (рис. 3).

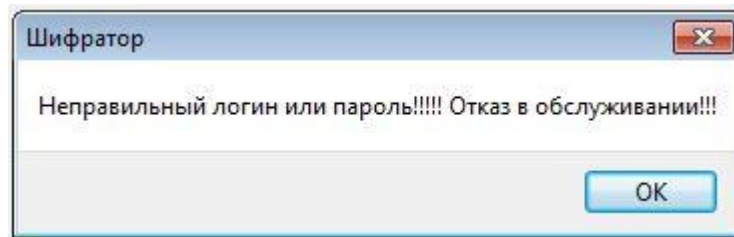


Рис.3 - Всплывающее сообщение.

9. Все окна должны быть подписаны (иметь названия или пояснения).
10. Интерфейс окна смены пароля или смены коэффициента при P должен иметь следующий вид (рис. 4).

A screenshot of a Windows-style window. The title bar is blue and contains the text 'Смена пароля (Коэффициента при P)' and standard window controls (minimize, maximize, close). The main area is light gray and contains three input fields. The first input field is preceded by the text 'Введите старый пароль(коэффициент при P)'. The second input field is preceded by the text 'Введите новый пароль(коэффициент при P)'. The third input field is preceded by the text 'Повторить новый пароль(коэффициент при P)'. At the bottom center, there is a button with the text 'Смена пароля (коэффициента)'.

Рис.4 - Окно смены пароля (коэффициента при P)

Требования по визуальному оформлению окна шифрования и расшифрования:

Разместить и оформить окно(окна) шифрования и расшифрования согласно рис.5. Окно должно иметь название «Шифрование/Дешифрование» и содержать следующие компоненты:

1. Компоненты, для указания пути к файлу с исходным текстом путь куда будет сохраняться файл с обработанным текстом.
2. Компоненты меню.

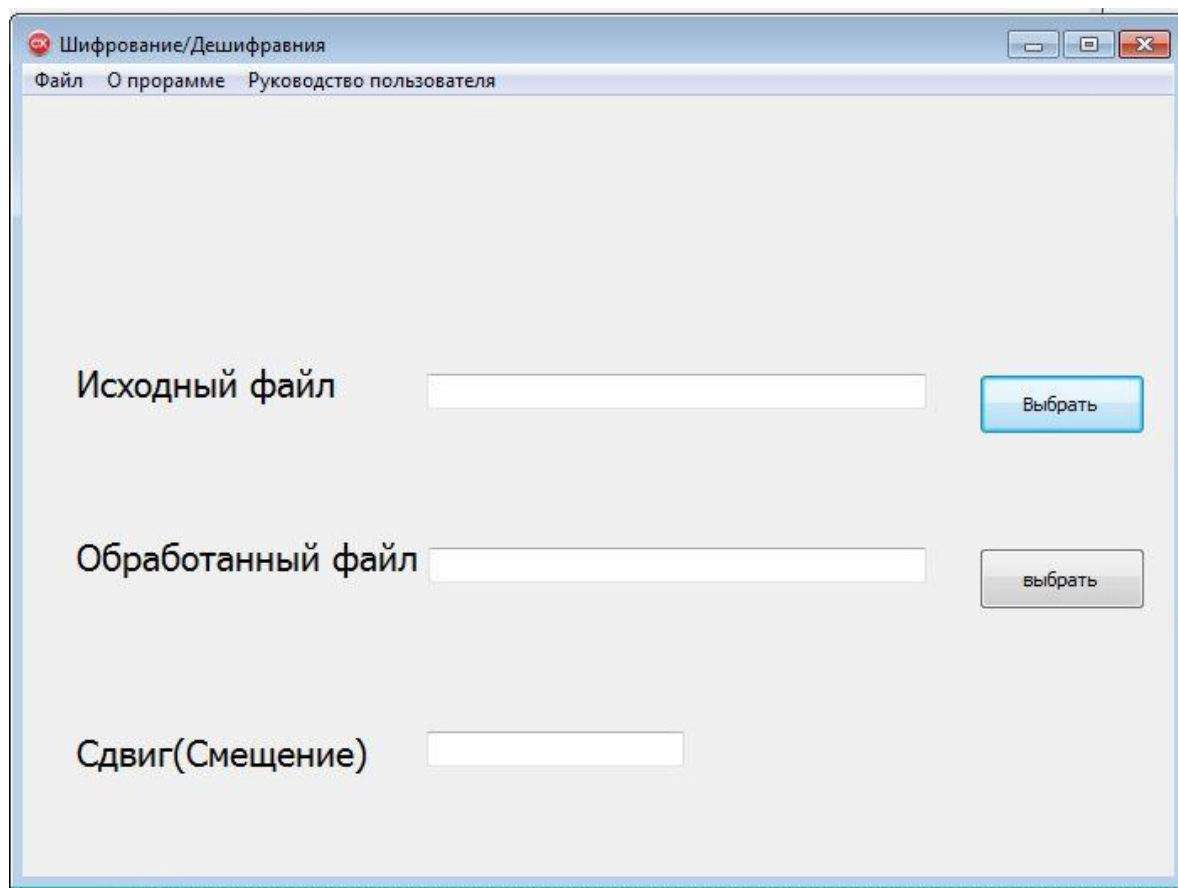


Рис.5 – главное окно шифратора

3. Окно шифрования (дешифрования) должно содержать главное меню (рис. 5) с пунктом «**Файл**» с подпунктами: «Открыть», «Сохранить как», «Выход», «Действие» которое содержит подпункты «Зашифровать», «Расшифровать» и пункты «**О программе**», «**Руководство пользователя**».

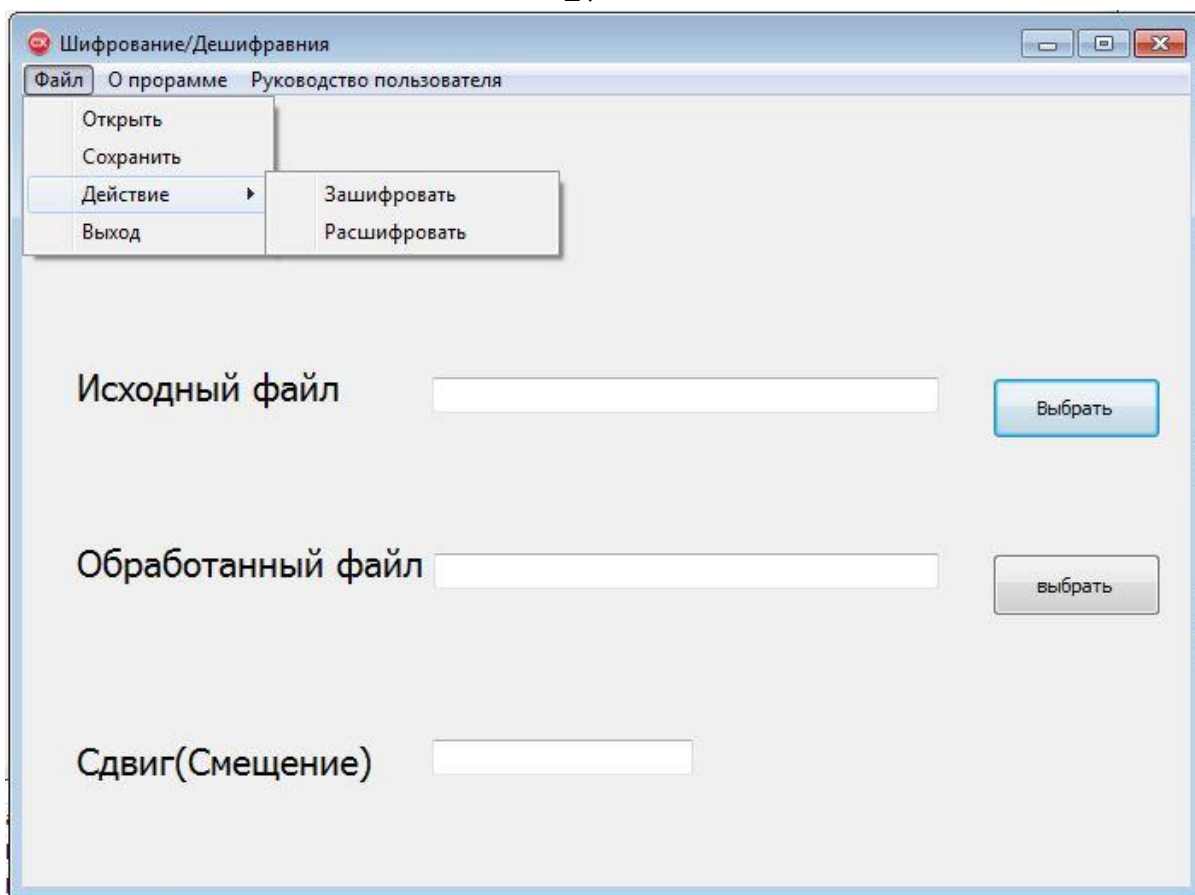


Рис. 5 – Меню окна

4. Пункт **«О программе»**, должен вызвать дополнительное модальное окно в котором должны содержаться сведения о названии программы, разработчике, текущей версии программы.

5. Пункт **«Руководство пользователя»**, должен содержать сведения порядке запуска программы, действиях пользователя во время работы с программой, перечень возможных неисправностей и способы их устранения.

6. Подпункт **«Выход»**, должен корректно завершать программу.

Преподаватель 34 кафедры (защиты информации
специальными методами и средствами)

А. Шандраков