

Документ подписан простой электронной подписью  
Информация о владельце:  
ФИО: Таскаев Сергей Владимирович  
Должность: Ректор  
Дата подписания: 15.09.2025 11:07:11  
Уникальный программный ключ:  
04c19ed8bfb98f3b6cb77a48809a878808522525



МИНОБРНАУКИ РОССИИ Федеральное государственное бюджетное образовательное учреждение высшего образования «Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ») Математический факультет Кафедра компьютерной безопасности и прикладной алгебры			
Фонд оценочных средств по практике «Учебная практика (учебно-лабораторный практикум)» по специальности 10.05.01 Компьютерная безопасность специализации № 1 «Анализ безопасности компьютерных систем»			
Версия документа - 1	стр. 1	Первый экземпляр _____	КОПИЯ № _____

**Фонд оценочных средств для промежуточной аттестации  
по практике  
Учебная практика (учебно-лабораторный практикум)**

Направление подготовки (специальность)  
10.05.01 Компьютерная безопасность

Направленность (профиль)  
специализация № 1 «Анализ безопасности компьютерных систем»

Присваиваемая квалификация  
специалист по защите информации

Форма обучения  
очная

Челябинск 2025 г.



МИНОБРНАУКИ РОССИИ  
Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ»)  
Математический факультет  
Кафедра компьютерной безопасности и прикладной алгебры

Фонд оценочных средств по практике «Учебная практика (учебно-лабораторный практикум)»  
по специальности 10.05.01 Компьютерная безопасность  
специализации № 1 «Анализ безопасности компьютерных систем»

Версия документа - 1

стр. 2

Первый экземпляр \_\_\_\_\_

КОПИЯ № \_\_\_\_\_

## Содержание

1. Паспорт фонда оценочных средств
2. Перечень формируемых компетенций
  - 2.1. Компетенции, закреплённые за дисциплиной
3. Содержание оценочных средств по дисциплине
  - 3.1. Виды оценочных средств
  - 3.2. Содержание оценочных средств
4. Порядок проведения и критерии оценивания промежуточной аттестации
  - 4.1. Порядок проведения промежуточной аттестации
  - 4.2. Критерии оценивания промежуточной аттестации по видам оценочных средств
  - 4.3. Результаты промежуточной аттестации и уровни сформированности компетенций



МИНОБРНАУКИ РОССИИ  
Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ»)  
Математический факультет  
Кафедра компьютерной безопасности и прикладной алгебры

Фонд оценочных средств по практике «Учебная практика (учебно-лабораторный практикум)»  
по специальности 10.05.01 Компьютерная безопасность  
специализации № 1 «Анализ безопасности компьютерных систем»

Версия документа - 1

стр. 3

Первый экземпляр \_\_\_\_\_

КОПИЯ № \_\_\_\_\_

## 1. ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ

Специальность 10.05.01 Компьютерная безопасность.

Специализация № 1 «Анализ безопасности компьютерных систем».

Наименование практики: **Учебно-лабораторный практикум.**

Вид практики: учебная.

Способы проведения учебной практики: стационарная.

Семестр (семестры) изучения: 2 семестр.

Форма (формы) промежуточной аттестации: зачет с оценкой (2 семестр).

## 2. ПЕРЕЧЕНЬ ФОРМИРУЕМЫХ КОМПЕТЕНЦИЙ

### 2.1. Компетенции, закреплённые за дисциплиной

Прохождение «Учебно-лабораторного практикума» направлено на формирование следующих компетенций:

Коды компетенции согласно ФГОС (ОПОП ВО)	Содержание компетенций согласно ФГОС (ОПОП ВО)	Индикаторы достижения компетенции согласно ОПОП	Перечень планируемых результатов обучения по дисциплине
1	2	3	4
ОПК-7	Способен создавать программы на языках высокого и низкого уровня, применять методы и инструментальные средства программирования для решения профессиональных задач, осуществлять обоснованный выбор инструментария	ОПК-7.1. Знает общие принципы построения, области и особенности применения языков программирования высокого и низкого уровня; язык программирования высокого и низкого уровня (объектно-ориентированное программирование); знает язык ассемблера персонального компьютера; базовые структуры данных; основные алгоритмы сортировки и поиска данных, комбинаторные и теоретико-графовые алгоритмы; общие сведения о методах проектирования, документирования, разработки, тестирования и отладки программного обеспечения. ОПК-7.2. Умеет работать с интегрированной средой разработки программного обеспечения; разрабатывать и реализовывать на языке высокого и низкого уровня	Знать: - основные понятия информатики и языков программирования; - язык программирования С и основы языка программирования С++. Уметь: - применять программные средства для решения математических задач; - использовать стандартные и сторонние библиотеки. Владеть: - навыками реализации базовых математических алгоритмов; - навыками работы в интегрированной среде разработки (IDE) MS Visual Studio.



МИНОБРНАУКИ РОССИИ  
Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ»)  
Математический факультет  
Кафедра компьютерной безопасности и прикладной алгебры

Фонд оценочных средств по практике «Учебная практика (учебно-лабораторный практикум)»  
по специальности 10.05.01 Компьютерная безопасность  
специализации № 1 «Анализ безопасности компьютерных систем»

Версия документа - 1

стр. 4

Первый экземпляр \_\_\_\_\_

КОПИЯ № \_\_\_\_\_

	программирования и способов организации программ	алгоритмы решения типовых профессиональных задач; применять известные методы программирования и возможности базового языка программирования для решения типовых профессиональных задач. ОПК-7.3. Владеет навыками разработки, документирования, тестирования и отладки программ; навыками разработки алгоритмов решения типовых профессиональных задач.	
ПК-3	Способен проводить анализ безопасности компьютерных систем	ПК-3.1. Обладает знаниями о уровнях защищенности и доверия в компьютерных системах; об оценках рисков, связанных с осуществлением угроз безопасности в отношении компьютерных систем; об оценках соответствия механизмов безопасности компьютерной системы требованиям существующих нормативных документов, а также их адекватности существующим рискам. ПК-3.2. Демонстрирует умения: проводить мониторинг, анализ и сравнение эффективности программно-аппаратных средств защиты информации в операционных системах; формулировать и разрабатывать предложения по устранению выявленных уязвимостей. ПК-3.3. Имеет практический опыт (навыки): выполнение анализа уязвимости компьютерных систем.	Знать: - базовые программные алгоритмы и структуры данных. Уметь: - применять базовые алгоритмы и структуры данных при решении прикладных задач. Владеть: - навыками реализации базовых алгоритмов.



МИНОБРНАУКИ РОССИИ  
Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ»)  
Математический факультет  
Кафедра компьютерной безопасности и прикладной алгебры

Фонд оценочных средств по практике «Учебная практика (учебно-лабораторный практикум)»  
по специальности 10.05.01 Компьютерная безопасность  
специализации № 1 «Анализ безопасности компьютерных систем»

Версия документа - 1

стр. 5

Первый экземпляр \_\_\_\_\_

КОПИЯ № \_\_\_\_\_

### 3. СОДЕРЖАНИЕ ОЦЕНОЧНЫХ СРЕДСТВ ПО ДИСЦИПЛИНЕ

#### 3.1. Виды оценочных средств

№ п/п	Код компетенции / планируемые результаты обучения	Контролируемые темы/ разделы	Наименование оценочного средства для текущего контроля	Наименование оценочного средства на промежуточной аттестации/№ задания
1.	ОПК-7, ПК-3	Основной этап: практический	Домашние и аудиторные задания	Отчет (практические задания)
2.	ОПК-7, ПК-3	Аналитический этап	Домашние и аудиторные задания	Отчет (практические задания)

Типовые задания, критерии и показатели оценивания в рамках текущего контроля представлены в рабочей программе дисциплины (модуля). Полные комплекты оценочных средств и контрольно-измерительных материалов хранятся на кафедре.



## 3.2. Содержание оценочных средств

### 3.2.1. Практические задания.

1. Реализовать функции преобразования из числа в строку и из строки в число.

1) Функция `my_atoi` (аналог стандартной функции `atoi`) должна принимать строку символов, представляющих число в  $p$ -ичной системе счисления ( $2 \leq p \leq 36$ ). Максимальный алфавит для  $p=36$  - это все цифры и все латинские буквы (и строчные и прописные).

```
int my_atoi (const char *str, int p);
```

Если передано  $p=0$ , то определить систему счисления из вида строки.

2) Функция `my_itoa` должна принимать число и переводить его в строку в  $p$ -ичной системе счисления.

```
void my_itoa (char *buf, int a, int p)
```

Написать программу, демонстрирующую работу этих функций.

Написать программу, принимающую в качестве аргумента имя файла и выводящую содержимое файла (каждый байт) в шестнадцатеричном виде.

2. Прочитать текст из файла (произвольного размера, произвольного содержимого), заменить вхождения одной подстроки на другую (замена текста), результат записать в другой файл.

Имена файлов и подстроки (исходная и заменяемая) получать из аргументов командной строки.

Если в тексте "adfkasddkjadfkdqwed" заменить "ad" на "12345", то получим "12345fkasddkj12345fkdaqwed".

Условие: в памяти нельзя хранить из файла данных одновременно больше чем 512 байт.

3. Написать программу генерации случайных паролей.

С помощью опций командной строки задаём следующее:

-m минимальная длина пароля

-M максимальная длина пароля

-n длина пароля

-a алфавит символов

-C[aADS] набор символов (указывается один или несколько символов из множества {a, A, D, S}), a - маленькие латинские символы, A - большие латинские символы, D - цифры, S - спецсимволы.

Проверять корректность опций и аргументов опций (одни и те же



опции не должны повторяться, в качестве длины должна передаваться числовая строка и т.д.). Опции могут передаваться в произвольном порядке. Могут встречаться другие опции, которые следует игнорировать. Если есть опция -m должна быть опция -M. Опции -m,-M и -n не совместимы вместе (выдавать соответствующее сообщение). Опции -a и -C не совместимы вместе (выдавать соответствующее сообщение).

```
generate.exe -m 10 -M 20 -CaD
```

```
generate.exe -n 10 -a 123456
```

4. Реализовать пузырьковую сортировку двумерного статического массива строк.

Одна символьная строка должна храниться в одной строке двумерного массива.

Строки должны сравниваться в лексикографическом порядке (функция сравнения строк strcmp).

Шаблон для задания предоставляется.

5. Реализовать арифметику больших чисел.

Большое число следует хранить в массиве 32-х битных машинных слов (unsigned int). В каждом слове (быть может, кроме последнего) должны храниться 4 байта большого числа. Бинарные алгебраические операции над большими числами должны реализовываться через последовательность действий над словами этих чисел.

Реализовать следующие операции:

- сложение и вычитание больших чисел;
- умножение и деление больших чисел;
- вывода большого числа;
- инициализации большого числа из строки шестнадцатеричных (десятичных по желанию) символов.

Шаблон для задания предоставляется.

6. Реализовать быструю сортировку строк, хранящихся в двусвязном списке.

Двусвязный список следует организовать через структуру ListEntry (хранящую ссылки на два соседних элемента), с помощью которой можно организовать список любых сущностей, не изменяя код работы со списком.

Поменять местами два элемента в сортируемом контейнере для списка должно означать "поменять местами два элемента списка" (а не смену



МИНОБРНАУКИ РОССИИ  
Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ»)  
Математический факультет  
Кафедра компьютерной безопасности и прикладной алгебры

Фонд оценочных средств по практике «Учебная практика (учебно-лабораторный практикум)»  
по специальности 10.05.01 Компьютерная безопасность  
специализации № 1 «Анализ безопасности компьютерных систем»

Версия документа - 1

стр. 8

Первый экземпляр \_\_\_\_\_

КОПИЯ № \_\_\_\_\_

указателей внутри элемента).

Шаблон для задания предоставляется.

7. Реализовать ассоциативный массив, позволяющий сопоставлять ключу (произвольной строке) значение (произвольную строку), на основе двоичного дерева поиска. Дерево должно поддерживать операции: вставку пары (ключ, значение), поиск значения по ключу, удаление узла дерева, очистка дерева.

Реализовать программу, демонстрирующую возможности ассоциативного массива.

Шаблон для задания предоставляется.

8. С использованием интерфейса WinAPI реализовать игру с графическим интерфейсом.

Игра может быть выбрана из списка, либо предложена самостоятельно (тогда необходимо согласовать с преподавателем).

Если это предусмотрено игрой, должна быть реализована функциональность ответных действий компьютера.

9. Дописать пример `examples\picture` и реализовать функции преобразования bmp-изображения:

- уменьшить яркость;
- увеличить яркость;
- преобразовать в черно-белое изображение;
- зеркально отразить по горизонтали, по вертикали;
- поворот вокруг центральной оси на 90 градусов (в обе стороны);
- уменьшить/увеличить изображение в указанное количество раз;

можно предложить две идеи: для уменьшения в  $n$  раз выбирать только каждый  $n$ -ый пиксель по строке и столбце, но лучше считать среднее арифметическое из квадрата  $n \times n$  пикселей в качестве значения нового пикселя, заменяющего этот квадрат.

- из одной существующей картинке получить картинку в 4 раза (по высоте и ширине в 2 раза) большую как объединение 4 копий исходной
- склеить несколько изображений в одно большее.

10. Реализовать арифметику многочленов с использованием статического полиморфизма.

Хранить многочлен в виде сортированного списка.



Реализовать следующие операции:

- вывод многочлена;
- инициализация;
- сложение ( $c = a+b$ ), добавление ( $a+=b$ );
- вычитание ( $-$ ), убавление ( $-=$ );
- умножение ( $*$ ), домножение ( $*=$ );
- деление ( $/$ ) (и нацело и с остатком),  $/=$ ;
- НОД;
- вычислить от конкретного значения ( $f(x) \rightarrow f(10)$ ).

Считаем, что всегда можно делить коэффициенты.

Придумать и реализовать тесты (сложные вычисления над многочленами).

11. Реализовать арифметику матриц с использованием динамического полиморфизма.

Реализовать следующие операции:

- вывод матриц;
- инициализация;
- сложение;
- вычитание;
- умножение;
- определитель;
- нахождение обратной матрицы.

Считаем, что всегда можно делить коэффициенты.

Придумать и реализовать тесты (сложные вычисления над матрицами).

12. Дописать пример `expr` для распознавания выражений с помощью регулярных выражений. Для больших чисел должна использоваться библиотека `gmp`.

1) Инициализация переменных константами, значением вычисления выражения и результатом выполнения функции:

```
var = 10;
```

```
var = var + 10;
```

2) арифметические выражения ( $+$ ,  $-$ ,  $*$ ,  $/$ ) над константами и переменными

```
10 + 10909090909090909987687697867;
```

```
var + 10;
```



```
var + var;
```

```
var = var1 + var2;
```

3) один уровень вложенности круглых скобок во всех возможных бинарных выражениях

```
(10 + 10) * 2
```

```
var = (var1 + var2) * (15 + 10)
```

4) задание функций с аргументами, локальными переменными и возвращаемым значением

```
function FunName(a)
```

```
    local var = 1;
```

```
    local b = a + 10;
```

```
    return var * b;
```

```
end FunName
```

```
var = FunName(15);    -> 25
```

13. Реализовать класс строк. Определить следующие функции и операции:

- конструктор по умолчанию;
- инициализирующие конструкторы;
- копирующий конструктор;
- деструктор;
- операции +, =, ==, +=, !=, <, > <=, >=, [], \*;
- операторы перенаправления в поток, из потока;
- получение char\*;

- реализовать совместное использование буфера для строк несколькими объектами с помощью механизма копирования при записи (для этого может понадобиться дополнительная абстракция).

Придумать и реализовать тесты.

14. На языке C реализовать простой редактор изображений.

Нельзя использовать сторонние библиотеки, только WinAPI.

Требования:

- 1) Поддержка форматов bmp, jpg, png (открытие, сохранение).
- 2) Выделение прямоугольных участков изображений с возможностью вырезания, копирования, вставки.
- 3) Рисование мышкой линий, прямых, прямоугольников с возможностью выбора цвета и толщины линии.
- 4) С помощью мышки увеличение/уменьшение размеров изображения.



## 15. На языке С реализовать шестнадцатеричный редактор файлов.

Нельзя использовать сторонние библиотеки, только WinAPI.

Необходимо реализовать компонент шестнадцатеричного редактора в виде отдельного окна, которое в качестве дочернего окна встраивается в приложение. Вся логика работы с редактированием шестнадцатеричного представления должна быть реализована в этом компоненте. Через оконные сообщения реализовать интерфейс, через который осуществляется настройка, загрузка данных в компонент, выгрузка отредактированных данных, уведомление о событиях.

Компонент реализуется полностью самостоятельно, нельзя использовать в качестве основы какие-то готовые окна. Символы можно отображать с помощью операции вывода текста на окне.

Данные отображаются в классическом виде: в каждой строке смещение, шестнадцатеричное представление байт, символьное представление байт (для которых это возможно).

Должна быть возможность указать компоненту фиксированное количество байт в строке и количество строк. Либо указать, что они высчитываются динамически из размера окна.

Должны поддерживаться функции:

- 1) Редактирование шестнадцатеричного и символьного представления.
- 2) Выделение участков шестнадцатеричного и символьного представления.
- 3) Операции вырезания, копирования, вставки (в шестнадцатеричном и символьном виде).
- 4) Поиск байт (в шестнадцатеричном и символьном виде).
- 5) Должна быть возможность работы с файлами произвольного размера. Т.е. полностью считывать файл в память нельзя.

## 16. На языке С реализовать текстовый редактор.

Нельзя использовать сторонние библиотеки, только WinAPI.

Необходимо реализовать компонент текстового редактора в виде отдельного окна, которое в качестве дочернего окна встраивается в приложение. Вся логика работы текстового редактора должна быть реализована в этом компоненте. Через оконные сообщения реализовать интерфейс, через который осуществляется настройка, загрузка данных в компонент, выгрузка отредактированных данных, уведомление о событиях.

Компонент реализуется полностью самостоятельно, нельзя



использовать в качестве основы какие-то готовые окна.

По сути требуется реализовать упрощённую версию стандартного окна edit.

Вывод графического представления (глифа) символа осуществляется самостоятельно (по сути, в виде отдельных пикселей) без использования системных функций вывода текста в окно. Т.е. надо сделать самостоятельный вывод некоторого своего шрифта.

Должны поддерживаться функции:

- 1) Отображение текста.
- 2) Редактирование (ввод, удаление, замена) текста.
- 3) Выделение текста клавиатурой и мышью.
- 4) Копирование, вставка, вырезание текста.
- 5) Поиск текста с подсвечиванием найденных фрагментов.

17. На языке C реализовать редактор электронных таблиц.

Нельзя использовать сторонние библиотеки, только WinAPI.

Необходимо реализовать компонент редактора в виде отдельного окна, которое в качестве дочернего окна встраивается в приложение. Вся логика работы редактора должна быть реализована в этом компоненте. Через оконные сообщения реализовать интерфейс, через который осуществляется настройка, загрузка данных в компонент, выгрузка отредактированных данных, уведомление о событиях.

Компонент реализуется полностью самостоятельно, нельзя использовать в качестве основы какие-то готовые окна. Символы можно отображать с помощью операции вывода текста на окне.

Должно поддерживаться открытие/сохранение файлов в формате csv и некотором своём формате (для реализации возможностей, не предусмотренных простым форматом csv).

Должны поддерживаться функции:

1. Предоставление традиционного интерфейса по отображению и редактированию электронных таблиц.
2. Возможность выделения ячеек клавиатурой и мышью.
3. Копирование, вставка, вырезание данных в ячейках.
4. Изменение размеров ячеек с помощью мышки.
5. Поддержка формулы сложения заданных (клавиатурой, мышью, перечислением) ячеек в качестве значения для ячейки.

18. Реализация простой базы данных и системы управления базой



данных (СУБД). Предоставлять по сети интерфейс взаимодействия. Поддерживать ограниченный набор SQL-команд.

Данные и метаданные хранить в файлах на диске. Для каждой таблицы создавать отдельный каталог. В файлах такого каталога хранить метаданные таблицы (типы, атрибуты названия колонок; число записей; индексы, ...) и данные (каждую строку можно хранить в отдельном файле).

Поддерживать SQL-команды:

1) создание таблицы

`CREATE TABLE table_name (col1_name col1_type col1_attr, col2_name col2_type2 col2_attr, ...)`

Поддерживать типы `int`, `char`, `varchar`.

Поддерживать атрибуты `NOT NULL`, `PRIMARY_KEY`, `UNIQUE`.

2) удаление таблицы

`DROP TABLE table_name`

3) запрос данных

`SELECT col1_name, col2_name, ... FROM table_name [WHERE condition]`

Поддерживать условия сравнения чисел и строк.

4) удаление данных

`DELETE FROM table_name [where condition]`

Поддерживать условия сравнения чисел и строк.

Поддерживать индекс по первичному ключу. Т.е. проверка условий на столбец, являющийся первичным ключом (может указываться при создании), должна происходить быстрее, чем для остальных столбцов. Для этого реализовать (хранимый на диске) ассоциативный массив, сопоставляющий ключу (значение в столбце) информацию о расположении строки с этим ключом в файле (имя файла или нужное смещение в файле). Обычно для этого используют деревья. Можно использовать отсортированный по ключам массив.

Придумать и реализовать некоторый протокол, по которому по сети клиенты смогут посылать команды и получать ответы.

Реализовать тестовую программу, которая по сети будет общаться с СУБД.

### 3.2.2. Темы для домашних и аудиторных заданий:

1. Синтаксис языка программирования C.
2. Синтаксис языка программирования C++.
3. Обработка массивов данных.



МИНОБРНАУКИ РОССИИ  
Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ»)  
Математический факультет  
Кафедра компьютерной безопасности и прикладной алгебры

Фонд оценочных средств по практике «Учебная практика (учебно-лабораторный практикум)»  
по специальности 10.05.01 Компьютерная безопасность  
специализации № 1 «Анализ безопасности компьютерных систем»

Версия документа - 1

стр. 14

Первый экземпляр \_\_\_\_\_

КОПИЯ № \_\_\_\_\_

4. Обработка строк.
5. Обработка изображений.
6. WinAPI.
7. Алгоритмы и структуры данных.
8. Статический и динамический полиморфизм.
9. Регулярные выражения.

### **3.2.3. Примеры домашних и аудиторных заданий:**

1. Программа генерации паролей.
2. Программа обработки BMP-изображений.
3. Графическая интерактивная игра.
4. Реализация арифметики больших чисел.
5. Программа замены текста в файлах.
6. Преобразование числа в строка и обратно.
7. Реализация арифметики многочленов.
8. Реализация арифметики матриц.
9. Разбор и вычисление арифметических выражений.



МИНОБРНАУКИ РОССИИ  
Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ»)  
Математический факультет  
Кафедра компьютерной безопасности и прикладной алгебры

Фонд оценочных средств по практике «Учебная практика (учебно-лабораторный практикум)»  
по специальности 10.05.01 Компьютерная безопасность  
специализации № 1 «Анализ безопасности компьютерных систем»

Версия документа - 1

стр. 15

Первый экземпляр \_\_\_\_\_

КОПИЯ № \_\_\_\_\_

## **4. ПОРЯДОК ПРОВЕДЕНИЯ И КРИТЕРИИ ОЦЕНИВАНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ**

### **4.1. Порядок проведения промежуточной аттестации**

Промежуточная аттестация проводится в форме защиты отчета по выполненным заданиями текущей аттестации.

Во время защиты студентам задаются уточняющие вопросы по тематике выполненных заданий.

Итоговая оценка за практику выставляется по результатам выполнения практических заданий текущего контроля в зависимости от количества сданных заданий.

### **4.2. Критерии оценивания промежуточной аттестации по видам оценочных средств**

#### **4.2.1 Критерии оценивания практических заданий отчета, домашних и аудиторных заданий**

Выполнение заданий предполагает некоторую программную реализацию, к которой предъявляются обычные требования по качеству кода. Код должен быть удобочитаемым, хорошо структурированным, расширяемым, удобным в сопровождении, написанным в едином стиле. Должна быть проведена функциональная декомпозиция (на подпрограммы, модули, пакеты и т.д.), реализованы необходимые программные абстракции. Реализации алгоритмов должны быть логичными и понятными. В программах недопустимы ошибки, приводящие к неработоспособности кода для некоторых возможных случаев либо к аварийному некорректному завершению программы. В случае наличия указанных выше проблем преподаватель может потребовать их исправления и повторной сдачи задания. Для одного задания предоставляется не более трёх попыток сдачи.

При сдаче задания производится опрос по техническим деталям реализации и по теории, используемой при выполнении заданий. Неудовлетворительный ответ будет означать несамостоятельность выполнения задания, что влечёт запрет сдачи этого задания.

### **4.3. Результаты промежуточной аттестации и уровни сформированности компетенций**



МИНОБРНАУКИ РОССИИ  
Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ»)  
Математический факультет  
Кафедра компьютерной безопасности и прикладной алгебры

Фонд оценочных средств по практике «Учебная практика (учебно-лабораторный практикум)»  
по специальности 10.05.01 Компьютерная безопасность  
специализации № 1 «Анализ безопасности компьютерных систем»

Версия документа - 1

стр. 16

Первый экземпляр \_\_\_\_\_

КОПИЯ № \_\_\_\_\_

Промежуточная аттестация по учебной практике осуществляется в форме дифференцированного зачета (зачет с оценкой).

Обязательной формой отчетности по результатам прохождения практики является письменный отчет студента. Отчет должен содержать информацию о выполненных заданиях, обычно количество заданий составляет 12-15.

Объем отчета, как правило, составляет порядка 5-10 страниц. На оформление отчета студенту отводятся 2-3 дня в конце практики.

Оценка выставляется по результатам защиты отчета в зависимости от количества выполненных заданий следующим образом:

«Отлично» – выполнены все задания;

«Хорошо» – не выполнено не более двух заданий (среди выполненных обязательно есть игра);

«Удовлетворительно» – выполнено не менее половины заданий;

«Неудовлетворительно» – выполнено менее половины заданий.

Особенности проведения процедуры оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья обозначены в рабочей программе дисциплины (модуля).

Уровни сформированности компетенций определяется следующим образом:

Оценка	Отлично	Хорошо	Удовлетворительно	Неудовлетворительно
Уровень освоения проверяемых компетенций	высокий	средний	базовый	недостаточный

