

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Гаскаев Сергей Валерьевич
Должность: Ректор
Дата подписания: 15.09.2025 11:13:06
Уникальный программный ключ:
04c19ed8bf98f4b6cb77a486b9a878808572523



МИНОБРАЗОВАНИЯ И НАУКИ РОССИИ
Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ»)

Фонд оценочных средств для промежуточной аттестации по практике «Учебная практика (практика по программированию)» по направлению подготовки 01.03.02 «Прикладная математика и информатика» направленности «Прикладная математика и искусственный интеллект» ФГБОУ ВО «ЧелГУ»

стр. 1

Фонд оценочных средств для промежуточной аттестации по практике
Учебная практика
Практика по программированию

Направление подготовки (специальность)
01.03.02 «Прикладная математика и информатика»

Направленность (профиль)
«Прикладная математика и искусственный интеллект»

Присваиваемая квалификация
Бакалавр

Форма обучения
Очная

Челябинск, 2025 г.



Содержание

1. Паспорт фонда оценочных средств	3
2. Перечень формируемых компетенций	4
3. Содержание оценочных средств по практике.....	6
3.1. Виды оценочных средств	6
3.2. Содержание оценочных средств	7
4. Порядок проведения и критерии оценивания промежуточной аттестации	7
4.1. Порядок проведения промежуточной аттестации	11
4.2. Критерии оценивания промежуточной аттестации по видам оценочных средств	11
4.3. Результаты промежуточной аттестации и уровни сформированности компетенций.....	11



1. Паспорт фонда оценочных средств

Направление подготовки: 01.03.02 Прикладная математика и информатика.

Направленность (профиль): Прикладная математика и искусственный интеллект.

Наименование практики: Учебная практика (практика по программированию).

Семестры: 2.

Тип практики: учебная.

Вид практики: практика по программированию.

Способы проведения практики: стационарная.

Форма промежуточной аттестации: дифференцированный зачет во 2 семестре.

Для оценивания результатов обучения используется балльно-рейтинговая система.



2. Перечень формируемых компетенций

2.1. Компетенции, закрепленные за практикой

Прохождение практики «Учебная практика (практика по программированию)» 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 практике
ОПК-2	Способен использовать и адаптировать существующие математические методы и системы программирования для разработки и реализации алгоритмов решения прикладных задач	ОПК-2.1. Имеет представление о существующих базовых математических методах и системах программирования, применяемых для разработки и реализации алгоритмов решения прикладных задач. ОПК-2.2. Демонстрирует умение применять математические методы и системы программирования для решения прикладных задач. ОПК-2.3. Имеет навыки разработки и применения алгоритмических и программных решений.	Знать: принципы работы современных операционных систем, систем имитационного моделирования, информационных систем глобальных сетей. Уметь: использовать современные системы имитационного моделирования, создавать информационные системы глобальных сетей. Владеть: навыками имитационного моделирования.
ПК-2	Способен использовать базовые алгоритмы и средства проектирования программного обеспечения	ПК-2.1. Обладает знаниями о существующих типовых шаблонах проектирования программного обеспечения; о методах и средствах проектирования программного обеспечения, структур данных, баз данных. ПК-2.2. Демонстрирует умение: применять типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке программного обеспечения; применять методы и средства проектирования программного	Знать: способы описания алгоритмов, языки и инструментальные среды программирования, методы отладки и тестирования программ. Уметь: описать и обосновать разработанные алгоритмы и внешние спецификации, разработать программный код, составить тесты и выполнить



		обеспечения, баз данных, структур данных. ПК-2.3. Имеет практический опыт (навыки): применения стандартных алгоритмов при проектировании программного обеспечения; разработки алгоритмов решения задач в соответствии с поставленными условиями; использования методов и приемов алгоритмизации поставленных задач.	тестирование программы, составить самодокументирование программы. Владеть: навыками проверки правильности работы программы.
--	--	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------



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
1	ОПК-2 Знать: принципы работы современных операционных систем, систем имитационного моделирования, информационных систем глобальных сетей.	Сбор и анализ литературы по поставленной задаче (реализуется в форме практической подготовки).	Отчетная документация (отчет и презентация доклада).	Отчетная документация
2	Уметь: использовать современные системы имитационного моделирования, создавать информационные системы глобальных сетей. Владеть: навыками имитационного моделирования.	Выбор программной архитектуры решения. Составление плана работ (реализуется в форме практической подготовки).	Рабочее приложение, решающее поставленную задачу.	Исходные тексты программ (тестов), исполняемые файлы программ
3	ПК-2 Знать: способы описания алгоритмов, языки и инструментальные среды программирования, методы отладки и тестирования программ. Уметь: описать и обосновать разработанные алгоритмы и внешние спецификации,	Разработка (уточнение) алгоритмов решения задачи (реализуется в форме практической подготовки).	Текст исходного кода программы и созданных библиотек.	
4	разработать программный код, составить тесты и выполнить тестирование программы,	Кодирование задачи (реализуется в форме практической подготовки).		
5	составить самодокументирование программы. Владеть: навыками проверки правильности работы программы.	Отладка программного кода (реализуется в форме практической подготовки).		
6		Составление и прогонка тестирующих подпрограмм		



№ п/п	Код компетенции / планируемые результаты обучения	Контролируемые темы / разделы	Наименование оценочного средства для текущего контроля	Наименование оценочного средства на промежуточной аттестации
		(реализуется в форме практической подготовки).		
7		Подготовка отчета по практике (реализуется в форме практической подготовки).		

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

Промежуточная аттестация проводится в виде дифференцированного зачета во 2 семестре.

Обязательные требования по реализации заданий:

- Все задания выполнять на языке СИ(C++)
- Для получения оценок «отлично» или «хорошо» НЕОБХОДИМО разработать, обосновать, и реализовать дополнительную функциональность (сверх той, что требуется по условиям конкретной задачи)
- Придерживаться следующего порядка разработки (разработка ОБЯЗАТЕЛЬНО ведется с созданием соответствующих библиотек):
 - a. Анализ предметной области
 - b. Проектирование интерфейсов задачи (программного и пользовательского)
 - c. Выбор подходящих структур данных для реализации интерфейса
 - d. Выбор порядка реализации подпрограмм интерфейса
 - e. Реализация подпрограмм с одновременным их тестированием
 - f. Создание консольного приложения, максимально удобным (для пользователя) образом демонстрирующего ПОЛНУЮ функциональность сделанного имитатора (или игры).

Дополнительно:

Для тех, кто хочет попробовать себя в по-настоящему новых технологиях и протестировать свои навыки в освоении современных языков программирования, допускается написать решения своих задач на языках:

1. Rust (<http://www.rust-lang.org>)
2. Julia (<http://julialang.org>)

Решивший выбранную задачу на одном из этих языков получит +1.5 к оценке за практику.

Задания на практику:

1. Разработать и реализовать программу имитирующую работу машины Тьюринга.



- a. Входные данные и программу задавать во внешнем файле.
- b. Предусмотреть вывод результата работы программы на консоль и или файл
- c. Предусмотреть обработку и вывод информации об ошибках в тексте программы и ее выполнения (в файл и на консоль)
- d. Реализовать протоколирование работы интерпретатора (по выполнению пользовательской программы), с возможностью ее вывода на консоль и или файл

Литературу по описанию исполнителя брать из лекций или интернета.

Работоспособность решений проверить, написав программы сложения и умножения натуральных чисел, заданных в единичной системе.

Программа должна использовать только стандартные библиотеки и классы и не должна вызывать сторонние программы, библиотеки или системные компоненты.

2. Разработать и реализовать программу имитирующую работу машины Поста.

- a. Входные данные и программу задавать во внешнем файле.
- b. Предусмотреть вывод результата работы программы на консоль и или файл
- c. Предусмотреть обработку и вывод информации об ошибках в тексте программы и ее выполнения (в файл и на консоль)
- d. Реализовать протоколирование работы интерпретатора (по выполнению пользовательской программы), с возможностью ее вывода на консоль и или файл

Литературу по описанию исполнителя брать из лекций или интернета.

Работоспособность решений проверить, написав программы сложения и умножения натуральных чисел, заданных в единичной системе.

Программа должна использовать только стандартные библиотеки и классы и не должна вызывать сторонние программы, библиотеки или системные компоненты.

3. Разработать и реализовать программу интерпретатор, моделирующую выполнение Нормальных алгоритмов Маркова (НАМ).

- a. Входные данные и программу задавать во внешнем файле.
- b. Предусмотреть вывод результата работы программы на консоль и или файл
- c. Предусмотреть обработку и вывод информации об ошибках в тексте программы и ее выполнения (в файл и на консоль)
- d. Реализовать протоколирование работы интерпретатора (по выполнению пользовательской программы), с возможностью ее вывода на консоль и или файл

Литературу по описанию исполнителя брать из лекций или интернета.

Работоспособность решений проверить, написав программы сложения и умножения натуральных чисел, заданных в единичной системе.

Программа должна использовать только стандартные библиотеки и классы и не должна вызывать сторонние программы, библиотеки или системные компоненты.

4. Разработать и реализовать консольный вариант игры "Сапер".

Игрок задает свое имя, размер поля и количество мин - новую игру, либо загружает записанную им ранее из файла. После генерации поля процесс игры осуществляется с помощью команд, введенных с клавиатуры.

Набор команд ОБЯЗАТЕЛЬНЫЙ для реализации:

1. 'o' x y //открыть клетку с координатами x,y
2. 'm' x y //поставить мину с координатами x,y
3. 's' <имя файла> //сохранить текущее состояние игры в файл <имя файла>



4. 'l' <имя файла> //восстановить игру из файла <имя файла>

5. 'c' //показать таблицу результатов (должна содержать следующие поля: название игрока, размер поля, количество мин, количество ходов, затраченное на их открытие)

6. 'h' //показать справку по командам

7. 'q' //выход из программы

Состояния полей отображать подходящими символами и обновлять по мере их изменения.

Программа должна использовать только стандартные библиотеки и классы и не должна вызывать сторонние программы, библиотеки или системные компоненты.

5. Мини Excel. Реализовать простую электронную таблицу в виде программы, выполняющейся из командной строки.

Она должна уметь обрабатывать ячейки таблицы, как и более продвинутые аналоги, только с упрощенным синтаксисом выражений. Каждая ячейка может содержать:

- Ничего

- Неотрицательное целое число

- Текстовые строки, которые начинаются с символа '

- Строки-выражения, которые начинаются с символа '=' и могут содержать неотрицательные целые числа, ссылки на ячейки и простые арифметические выражения.

Скобки запрещены, у всех операций одинаковый приоритет. Поля содержащие выражения и числа не содержат пробельных символов.

Ссылки на ячейки состоят из одной заглавной латинской буквы и следующей за ней цифры.

Грамматика ячейки:

expression ::= '=' term {operation}

term ::= cell_reference | nonnegative_number

cell_reference ::= [A-Z][1-9]

operation ::= '+' | '-' | '*' | '/'

text ::= "" {любая последовательность печатных символов}

Процесс обработки:

- Все выражения должны быть заменены на вычисленный результат.

- Все вычисления выполняются с помощью целочисленной арифметики со знаком.

- Ячейки с текстом должны быть вычислены как соответствующий текст без префикса '.

- Операции над строками текста запрещены.

- В случае любой ошибки вычисления формулы, вычисляемая ячейка должна содержать слово-сообщение об ошибке, начинающееся с символа '#'. Используйте короткие, ясные сообщения. Не надо предоставлять подробности об ошибках в выводе.

Программа должна использовать только стандартные библиотеки и классы и не должна вызывать сторонние программы, библиотеки или системные компоненты.

Ввод и вывод

– Программа получает описание таблицы с формулами из стандартного ввода, вычисляет ее и печатает полученный результат в стандартный вывод.

– Входные данные представлены таблицей, элементы строк которой разделены табуляциями.

– Первая строка содержит пару чисел, разделенных табуляцией - высоту и ширину таблицы, соответственно. Затем идут строки с ячейками таблицы, в грамматике, приведенной выше.

– Выходные данные должны содержать только ожидаемую информацию, включая сообщения об ошибках, и никакой другой информации в выводе не должно быть



- Выходные данные должны быть отформатированы в соответствии с приведенным ниже примером.
- Программа должна предусмотреть сохранение результата в задаваемый пользователем файл.

Пример входных данных:

```
3          4
12      =C2  3      'Sample
=A1+B1*C1/5 =A2*B1  =B3-C3 'Spread
'Test =4-3 5 'Sheet
```

Ожидаемый вывод на консоль:

```
12  -4  3  Sample
4   -16 -4  Spread
Test 1   5  Sheet
```



4. Порядок проведения и критерии оценивания промежуточной аттестации

4.1. Порядок проведения промежуточной аттестации

В последний день учебной практики проводится дифференцированный зачет.

Результирующая оценка ставится по результатам публичной защиты результатов проделанной работы на основе предоставленных студентом:

1. работающей версии программы, выполняющей поставленную задачу;
2. текста исходного кода программы (включая созданные студентом библиотеки);
3. бумажной и электронной версий отчета о проделанной работе, сделанного согласно установленным требованиям (см. соотв. раздел РПД) и сданного не позднее чем за день до проведения промежуточной аттестации (защиты).

4.2. Критерии оценивания промежуточной аттестации по видам оценочных средств

Максимальная стоимость оценочных средств:

Рабочее приложение, решающее поставленную задачу – 40 баллов.

Текст исходного кода программы и созданных библиотек – 30 баллов.

Отчетная документация – 30 баллов.

Итого: 100 баллов.

Примечание. Следующие недостатки работы считаются грубыми: при наличии одного из них ставится максимальная оценка – «удовлетворительно»:

1. Ошибка при запуске программы (запуск осуществляется на компьютере удовлетворяющим требованиям для работы программы и в соответствии с указаниями, описанными в отчете).
2. Ошибки работы программы на тестовых данных, указанных в отчете (документации).
3. Ошибки компиляции программы из исходных текстов.
4. Отсутствие на момент защиты одного из пунктов отчетности.
5. Отсутствие студента на защите.

4.3. Результаты промежуточной аттестации и уровни сформированности компетенций

Балльно-рейтинговая система оценки работы студента по результатам прохождения практики выстраивается на основе балловой оценки, предоставляемой студентом отчетности. Оценка «отлично» выставляется за 91-100 баллов, «хорошо» - за 71-90 баллов, «удовлетворительно» за 50-70 баллов.

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:

1. Продвинутый уровень сформированности компетенций соответствует оценке «отлично»:

Обучающийся владеет знаниями предмета в полном объеме учебной программы, достаточно глубоко осмысливает дисциплину; самостоятельно, в логической последовательности и исчерпывающе отвечает на все вопросы, подчеркивает при этом самое существенное, умеет анализировать, сравнивать, классифицировать, обобщать, конкретизировать и систематизировать изученный материал, выделять в нем главное; устанавливать причинно-следственные связи; четко формирует ответы.

2. Базовый уровень соответствует оценке «хорошо»:

Обучающийся владеет знаниями дисциплины почти в полном объеме программы (имеются пробелы знаний только в некоторых, особенно сложных разделах); самостоятельно и отчасти при



наводящих вопросах дает полноценные ответы на вопросы; не всегда выделяет наиболее существенное, не допускает вместе с тем серьезных ошибок в ответах.

3. Пороговый уровень соответствует оценке «удовлетворительно»:

Обучающийся владеет основным объемом знаний по дисциплине; проявляет затруднения в самостоятельных ответах, оперирует неточными формулировками; в процессе ответов допускает ошибки по существу вопросов.

4. Низкий уровень соответствует оценке «неудовлетворительно»:

Обучающийся не освоил обязательного минимума знаний предмета, не способен ответить на вопросы даже при дополнительных наводящих вопросах экзаменатора.

