

Документ подписан простой электронной подписью  
Информация о владельце:  
ФИО: Гаскаев Сергей Валерьевич  
Должность: Ректор  
Дата подписания: 15.06.2026 12:22:45  
Уникальный программный ключ:  
04c19ed8bf998f3b6c77348c9ca8788b372473



МИНОБРНАУКИ 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 02.03.02 «Фундаментальная информатика и информационные технологии» направленности «Прикладное программирование и системы искусственного интеллекта» ФГБОУ ВО «ЧелГУ»

стр. 1

**Фонд оценочных средств для промежуточной аттестации по практике  
«Учебная практика (практика по программированию)»**

Направление подготовки (специальность)  
**02.03.02 «Фундаментальная информатика и информационные технологии»**

Направленность (профиль)  
**«Прикладное программирование и системы искусственного интеллекта»**

Присваиваемая квалификация  
**Бакалавр**

Форма обучения  
**Очная**

Год набора  
**2026**

Челябинск, 2026 г.



## Содержание

1. Паспорт фонда оценочных средств .....	3
2. Перечень формируемых компетенций .....	4
3. Содержание оценочных средств по дисциплине .....	6
3.1. Виды оценочных средств .....	6
3.2. Содержание оценочных средств .....	8
4. Порядок проведения и критерии оценивания промежуточной аттестации .....	12
4.1. Порядок проведения промежуточной аттестации .....	12
4.2. Критерии оценивания промежуточной аттестации по видам оценочных средств .....	12
4.3. Результаты промежуточной аттестации и уровни сформированности компетенций.....	12



МИНОБРНАУКИ 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 02.03.02 «Фундаментальная информатика и информационные технологии» направленности «Прикладное программирование и системы искусственного интеллекта» ФГБОУ ВО «ЧелГУ»

стр. 3

## 1. Паспорт фонда оценочных средств

Направление подготовки: 02.03.02 Фундаментальная информатика и информационные технологии.

Направленность (профиль): Прикладное программирование и системы искусственного интеллекта.

Практика: Учебная практика (практика по программированию).

Семестры: 2.

Форма промежуточной аттестации: дифференцированный зачет во 2 семестре.

Для оценивания результатов обучения используется балльно-рейтинговая система.



## 2. Перечень формируемых компетенций

Прохождение практики «Учебная практика (практика по программированию)» направлено на формирование компетенций, приведённых в 1.

Таблица 1. Результаты обучения по дисциплине.

Код и наименование компетенции согласно ФГОС	Индикаторы достижения компетенций согласно ОПОП ВО	Перечень планируемых результатов обучения по дисциплине
1	2	3
<b>ОПК-2</b> Способен применять компьютерные/ суперкомпьютерные методы, современное программное обеспечение, в том числе отечественного происхождения, для решения задач профессиональной деятельности	ОПК-2.1. Демонстрирует знание методов использования инструментальных средств, готового программного обеспечения и библиотек; знаком с содержанием Единого реестра российских программ. ОПК-2.2. Демонстрирует умения выбирать и использовать инструментальные средства, готовое программное обеспечение и библиотеки. ОПК-2.3. Имеет практический опыт решения задач анализа, интеграции различных типов программного обеспечения и сетевых коммуникаций.	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 навыком имитационного моделирования.
<b>ОПК-3</b> Способен к разработке алгоритмических и программных решений в области системного и прикладного программирования, математических, информационных и имитационных моделей, созданию информационных ресурсов глобальных сетей, образовательного контента, прикладных баз данных, тестов и средств тестирования систем и средств на соответствие стандартам и исходным требованиям	ОПК-3.1. Демонстрирует знание теории алгоритмов, методологии и технологии программирования, основные принципы построения математических, информационных и имитационных моделей. ОПК-3.2. Способен разрабатывать алгоритмические и программные решения, создавать информационные ресурсы на базе готовых решений. ОПК-3.3. Имеет практический опыт использования технологий разработки программного обеспечения.	Знать современные языки программирования, современные библиотеки и пакеты программ. Уметь создавать программный код с использованием современных языков программирования и библиотек. Владеть навыком программирования.
<b>ПК-1</b> Способен проводить под научным руководством локальные научно-исследовательские и опытно-конструкторские разработки на основе существующих методов в конкретной области профессиональной деятельности.	ПК-1.1. Обладает знаниями о методологии и этапах выполнения научно-исследовательской работы; о методах решения научных задач; о методике подготовки отчета, в том числе выпускной квалификационной работы. ПК-1.2. Демонстрирует умения: обрабатывать и анализировать научно-техническую информацию и результаты исследований; выполнять под научным руководством научно-исследовательскую или опытно-конструкторскую разработку в	Знать структуру отчета по практике. Уметь выполнить анализ поставленной задачи. Владеть навыком подготовки библиографии по тематике решаемой задачи.



	<p>конкретной области профессиональной деятельности.</p> <p>ПК-1.3. Имеет практический опыт (навыки): научной аргументации при анализе объекта научной и профессиональной деятельности; подготовки научных обзоров, публикаций, рефератов и библиографий по тематике проводимых исследований.</p>	
<p><b>ПК-2</b> Способен к осуществлению интеграции программных модулей и компонент и проверки работоспособности программного продукта на основе международных и профессиональных стандартов информационных технологий, современных парадигм и методологий, инструментальных и вычислительных средств, методов и механизмов оценки и анализа функционирования средств и систем информационных технологий.</p>	<p>ПК-2.1. Обладает знаниями о методах и средствах сборки модулей и компонент программного обеспечения, о разработке процедур для развертывания программного обеспечения, миграции и преобразования данных, о создании программных интерфейсов; о методах и механизмах оценки и анализа функционирования средств и систем информационных технологий; о международных и профессиональных стандартах информационных технологий, о современных парадигмах и методологиях, инструментальных и вычислительных средствах.</p> <p>ПК-2.2. Демонстрирует умения: применять методы и средства сборки модулей и компонент программного обеспечения, разработки процедур для развертывания программного обеспечения, миграции и преобразования данных, создания программных интерфейсов; проводить проверку и оценку работоспособности программного продукта.</p> <p>ПК-2.3. Имеет практический опыт (навыки): сборки модулей и компонент программного обеспечения, разработки процедур для развертывания программного обеспечения, миграции и преобразования данных, создания программных интерфейсов; оценки работоспособности программного продукта.</p>	<p>Знать способы описания алгоритмов, языки и инструментальные среды программирования, методы отладки и тестирования программ.</p> <p>Уметь описать и обосновать разработанные алгоритмы и внешние спецификации, разработать программный код, составить тесты и выполнить тестирование программы, составить самодокументирование программы.</p> <p>Владеть навыком проверки правильности работы программы.</p>
<p><b>ПК-3</b> Способен к разработке требований и проектированию программного обеспечения на основе применения базовых математических знаний и информационных технологий при решении проектно-технических и прикладных задач.</p>	<p>ПК-3.1. Обладает знаниями о методах и средствах проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов.</p> <p>ПК-3.2. Демонстрирует умения: разрабатывать требования к программному продукту, применять методы и средства проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов.</p> <p>ПК-3.3. Имеет практический опыт (навыки) проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов.</p>	<p>Знать методы и средства проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов.</p> <p>Уметь разрабатывать требования к программному продукту, применять методы и средства проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов.</p> <p>Владеть навыком проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов.</p>



### 3. Содержание оценочных средств по дисциплине

#### 3.1. Виды оценочных средств

Таблица 2. Виды оценочных средств.

Код, наименование компетенции согласно ФГОС	Перечень планируемых результатов обучения по дисциплине	Контролируемые темы/разделы (номер и название раздела из РПД п.2.2)	Семестр	Номер задания	Наименование оценочного средства
<b>ОПК-2</b> Способен применять компьютерные/ суперкомпьютерные методы, современное программное обеспечение, в том числе отечественного происхождения, для решения задач профессиональной деятельности	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)	2		Отчетная документация (отчет и презентация доклада).  Рабочее приложение, решающее поставленную задачу.  Текст исходного кода программы и созданных библиотек.
	Уметь использовать современные системы имитационного моделирования, создавать информационные системы глобальных сетей.	Выбор программной архитектуры решения. Составление плана работ (реализуется в форме практической подготовки)			
	Владеть навыком имитационного моделирования.	Разработка (уточнение) алгоритмов решения задачи (реализуется в форме практической подготовки)			
<b>ОПК-3</b> Способен к разработке алгоритмических и программных решений в области системного и прикладного программирования, математических, информационных и имитационных моделей, созданию информационных ресурсов глобальных сетей, образовательного контента, прикладных баз данных, тестов и средств тестирования систем и средств на соответствие стандартам и исходным требованиям	Знать современные языки программирования, современные библиотеки и пакеты программ.	Кодирование задачи (реализуется в форме практической подготовки)			
	Уметь создавать программный код с использованием современных языков программирования и библиотек.	Отладка программного кода (реализуется в форме практической подготовки)			
	Владеть навыком программирования.	Составление и прогонка тестирующих подпрограмм (реализуется в форме практической подготовки)  Подготовка отчета по практике (реализуется в форме практической подготовки)			



<b>ПК-1</b> Способен проводить под научным руководством локальные научно-исследовательские и опытно-конструкторские разработки на основе существующих методов в конкретной области профессиональной деятельности.	Знать структуру отчета по практике.				
	Уметь выполнить анализ поставленной задачи.				
	Владеть навыком подготовки библиографии по тематике решаемой задачи.				
<b>ПК-2</b> Способен к осуществлению интеграции программных модулей и компонент и проверки работоспособности и программного продукта на основе международных и профессиональных стандартов информационных технологий, современных парадигм и методологий, инструментальных и вычислительных средств, методов и механизмов оценки и анализа функционирования средств и систем информационных технологий.	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 тестирование программы, составить самодокументирование программы.				
	Владеть навыком проверки правильности работы программы.				
<b>ПК-3</b> Способен к разработке требований и проектированию программного обеспечения на основе применения базовых математических знаний и информационных	Знать методы и средства проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов.				
	Уметь разрабатывать требования к программному продукту, применять методы и средства проектирования				



технологий при решении проектно-технических и прикладных задач.	программного обеспечения, структур данных, баз данных, программных интерфейсов.				
	Владеть навыком проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов.				

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



с. 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

с. 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

с. 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 //пометить клетку 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

п

р

о

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

в

Максимальная стоимость оценочных средств:

е

Рабочее приложение, решающее поставленную задачу – 30 баллов.

д

Текст исходного кода программы и созданных библиотек – 25 баллов.

е

Отчетная документация – 25 баллов.

н

Защита – 20 баллов.

и

Итого: 100 баллов.

я

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

п

1. Ошибка при запуске программы (запуск осуществляется на компьютере удовлетворяющим требованиям для работы программы и в соответствии с указаниями, описанными в отчете).

р

2. Ошибки работы программы на тестовых данных, указанных в отчете (документации).

о

3. Ошибки компиляции программы из исходных текстов.

м

4. Отсутствие на момент защиты одного из пунктов отчетности.

е

5. Отсутствие студента на защите.

ж

у

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

т

о

ч

в

ы

с

т

р

а

н

н

о

т

е

р

а

н

и

и

и

и

и

и

и



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.

