

МИНОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Институт прикладной математики и компьютерных наук



Оценочные материалы текущего контроля и промежуточной аттестации по учебной
практике
(Оценочные средства по дисциплине)

Объектно-ориентированное программирование

по направлению подготовки

01.03.02 Прикладная математика и информатика

Направленность (профиль) подготовки:

Математическое моделирование и информационные системы

ОС составила:

канд. физ.-мат. наук, доцент

доцент кафедры компьютерной безопасности

 - Е.Г. Пахомова

Рецензент:

Заведующий кафедрой компьютерной безопасности,

канд. техн. наук, доцент

 С.А. Останин

Оценочные средства одобрены на заседании учебно-методической комиссии института прикладной математики и компьютерных наук (УМК ИПМКН).

Протокол от 08.06.2023 г. № 02

Председатель УМК ИПМКН,

д-р техн. наук, профессор

 С.П. Сущенко

Оценочные средства (ОС) являются элементом системы оценивания сформированности компетенций у обучающихся в целом или на определенном этапе ее формирования.

ОС разрабатывается в соответствии с рабочей программой (РП) практики.

1. Компетенции и результаты обучения, формируемые в результате освоения дисциплины

Компетенция	Индикатор компетенции	Код и наименование результатов обучения (планируемые результаты обучения, характеризующие этапы формирования компетенций)	Критерии оценивания результатов обучения			
			Отлично	Хорошо	Удовлетворительно	Неудовлетворительно
ОПК-2. Способен использовать и адаптировать существующие математические методы и системы программирования для разработки и реализации алгоритмов решения прикладных задач.	ИОПК-2.1. Обладает навыками объектно-ориентированного программирования для решения прикладных задач в профессиональной деятельности.	ОР-ОПК-2.1. Обучающийся сможет: - применять объектно-ориентированное программирование для решения задач в профессиональной деятельности	Уверенно применяет объектно-ориентированное программирование для решения задач в профессиональной деятельности	Применяет объектно-ориентированное программирование для решения задач в профессиональной деятельности	Не уверенно применяет объектно-ориентированное программирование для решения задач в профессиональной деятельности	Не может применять объектно-ориентированное программирование для решения задач в профессиональной деятельности
	ИОПК-2.2. Проявляет навыки использования основных языков программирования, основных методов разработки программ, стандартов оформления программной документации.	ОР-ОПК-2.2. Обучающийся сможет: - использовать основные методы разработки программ - оформлять программную документацию	- Уверенно использует основные методы разработки программ. - Грамотно оформляет программную документацию	- Использует основные методы разработки программ. - Умеет оформлять программную документацию	- Не уверенно использует основные методы разработки программ. - Допускает ошибки при оформлении программной документации	- Не может использовать основные методы разработки программ. - Не может грамотно оформлять программную документацию
	ИОПК-2.3. Демонстрирует умение отбора среди существующих математических методов, наиболее подходящих для решения конкретной прикладной задачи.	ОР-ОПК-2.3. Обучающийся сможет: - анализировать поставленную прикладную задачу - подбирать для ее решения подходящие математические методы	- Проводит всесторонний анализ поставленной прикладной задачи - подбирает для ее решения оптимальные математические методы	- Проводит анализ поставленной прикладной задачи - подбирает для ее решения достаточно оптимальные математические методы	- Анализ поставленной прикладной задачи является неполным; - предложенный для ее решения метод, не является оптимальным	- Не может провести анализ поставленной прикладной задачи - Не может подобрать для ее решения подходящие математические методы

	ИОПК-2.4. Демонстрирует умение адаптировать существующие математические методы для решения конкретной прикладной задачи.	ОР-ОПК-2.4. Обучающийся сможет: - адаптировать существующие математические методы для решения конкретной прикладной задачи	Уверенно адаптирует существующие математические методы для решения конкретной прикладной задачи	Адаптирует существующие математические методы для решения конкретной прикладной задачи	Не уверенно адаптирует существующие математические методы для решения конкретной прикладной задачи	Не может адаптировать существующие математические методы для решения конкретной прикладной задачи
ОПК-4. Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности.	ИОПК-4.1. Обладает необходимыми знаниями в области информационных технологий, в том числе понимает принципы их работы.	ОР-ОПК-4.1. Обучающийся сможет: - понимать принципы работы современных информационных технологий	В полной мере понимает принципы работы современных информационных технологий	Понимает принципы работы современных информационных технологий	Не вполне понимает принципы работы современных информационных технологий	Не понимает принципы работы современных информационных технологий
	ИОПК-4.2. Применяет знания, полученные в области информационных технологий, при решении задач профессиональной деятельности.	ОР- ОПК-4.2. Обучающийся сможет: - применять знания, полученные в области информационных технологий, для решения задач профессиональной деятельности	Грамотно и свободно применяет знания, полученные в области информационных технологий для решения задач профессиональной деятельности	Применяет знания, полученные в области информационных технологий для решения задач профессиональной деятельности	Не уверенно применяет знания, полученные в области информационных технологий для решения задач профессиональной деятельности	Не может применять знания, полученные в области информационных технологий для решения задач профессиональной деятельности
	ИОПК-4.3. Использует современные информационные технологии на всех этапах решения задач профессиональной деятельности.	ОР-ОПК-4.3. Обучающийся сможет: - применять современные информационные технологии на всех этапах профессиональной деятельности	Уверенно применяет современные информационные технологии на всех этапах профессиональной деятельности	Применяет современные информационные технологии на всех этапах профессиональной деятельности	Не уверенно применяет современные информационные технологии на некоторых этапах профессиональной деятельности	Не может применять современные информационные технологии в профессиональной деятельности

	<p>ИОПК-4.4. Демонстрирует умение составлять научные обзоры, рефераты и библиографии по тематике научных исследований.</p>	<p>ОР-ОПК-4.4. Обучающийся сможет:</p> <ul style="list-style-type: none"> - составлять научные обзоры, рефераты и делать доклады по теме исследования - правильно цитировать и делать ссылки на используемые источники в письменных работах 	<ul style="list-style-type: none"> - Грамотно составляет научные обзоры, рефераты и делать доклады по теме исследования; - правильно цитирует и делает ссылки на используемые источники в письменных работах 	<ul style="list-style-type: none"> - Составляет научные обзоры, рефераты и делать доклады по теме исследования; - цитирует и делает ссылки на используемые источники в письменных работах 	<ul style="list-style-type: none"> - Способен составлять научные обзоры, рефераты и делать доклады по теме исследования, но проработка источников не достаточно полная; - допускает ошибки при цитировании и оформлении ссылок на используемые источники в письменных работах 	<ul style="list-style-type: none"> - Не может грамотно составлять научные обзоры, рефераты и делать доклады по теме исследования; - не умеет правильно цитировать и делать ссылки на используемые источники в письменных работах
<p>ПК-2. Способность формализовать и алгоритмизировать поставленную задачу, написать программный код, а также верифицировать работоспособность обеспечения и исправить дефекты</p>	<p>ИПК-2.1. Осуществляет построение формальной модели и алгоритма для поставленной задачи, написание программного кода с использованием языков программирования, верификацию работоспособности программного обеспечения и исправление дефектов.</p>	<p>ОР-ПК-2.1. Обучающийся сможет:</p> <ul style="list-style-type: none"> - строить формальную модель для поставленной задачи; - писать программный код; - верифицировать работоспособность программного обеспечения и исправлять дефекты 	<ul style="list-style-type: none"> - Уверенно строит формальную модель для поставленной задачи; - грамотно пишет программный код; - уверенно верифицирует работоспособность программного обеспечения и исправляет дефекты 	<ul style="list-style-type: none"> - Способен строить формальную модель для поставленной задачи; - способен писать программный код; - способен верифицировать работоспособность программного обеспечения и исправлять дефекты 	<ul style="list-style-type: none"> - Способен строить формальную модель для большинства поставленных задач; - в большинстве случаев способен писать программный код; - в большинстве случаев способен верифицировать работоспособность программного обеспечения и исправлять дефекты 	<ul style="list-style-type: none"> - Не способен строить формальную модель для поставленной задачи; - не способен писать программный код; - не способен верифицировать работоспособность программного обеспечения и исправлять дефекты

	<p>ИПК-2.2. Осуществляет оформление программного кода в соответствии с установленными требованиями, разработку процедур верификации работоспособности и измерения характеристик программного обеспечения, разработку тестовых наборов данных.</p>	<p>ОР-ПК-2.2. Обучающийся сможет:</p> <ul style="list-style-type: none"> - оформлять программный код в соответствии с установленными требованиями; - разрабатывать процедуры верификации работоспособности программного обеспечения; - разрабатывать наборы тестовых данных 	<ul style="list-style-type: none"> - Грамотно оформляет программный код в соответствии с установленными требованиями; - уверенно разрабатывает процедуры верификации работоспособности программного обеспечения; - уверенно разрабатывает наборы тестовых данных 	<ul style="list-style-type: none"> - Способен оформлять программный код в соответствии с установленными требованиями; - способен разрабатывать процедуры верификации работоспособности программного обеспечения; - способен разрабатывать наборы тестовых данных 	<ul style="list-style-type: none"> - В большинстве случаев способен оформлять программный код в соответствии с установленными требованиями; - в большинстве случаев способен разрабатывать процедуры верификации работоспособности программного обеспечения; - в большинстве случаев способен разрабатывать наборы тестовых данных 	<ul style="list-style-type: none"> - не способен оформлять программный код в соответствии с установленными требованиями; - не способен разрабатывать процедуры верификации работоспособности программного обеспечения; - не способен разрабатывать наборы тестовых данных
	<p>ИПК-2.3. Осуществляет работу с системой контроля версий, рефакторинг и оптимизацию программного кода.</p>	<p>ОР-ПК2-2.3. Обучающийся сможет:</p> <ul style="list-style-type: none"> - осуществлять работу с системой контроля версий; - осуществлять рефакторинг и оптимизацию программного кода 	<ul style="list-style-type: none"> - Уверенно осуществляет работу с системой контроля версий; - уверенно осуществляет рефакторинг и оптимизацию программного кода 	<ul style="list-style-type: none"> - Способен осуществлять работу с системой контроля версий; - способен осуществлять рефакторинг и оптимизацию программного кода 	<ul style="list-style-type: none"> - В большинстве случаев способен осуществлять работу с системой контроля версий; - в большинстве случаев способен осуществлять рефакторинг и оптимизацию программного кода 	<ul style="list-style-type: none"> - Не способен осуществлять работу с системой контроля версий; - не способен осуществлять рефакторинг и оптимизацию программного кода

ПК-3 Способность формализовывать, согласовывать и документировать требования к системе и подсистеме, обрабатывать запросы на изменение требований к системе и подсистеме, выявлять и формализовывать риски, анализировать проблемные ситуации.	ИПК-3.1. Реализовывает построение формализованной математической модели системы (подсистемы), введение целевой функции системы, подсистемы и ограничений, соответствующих требованиям к системе (подсистеме).	ОР-ПК-3.1. Обучающийся сможет: - реализовывать построение формализованной математической модели системы (подсистемы); - вводить целевые функции системы (подсистемы), соответствующие ограничениям	- Уверенно реализует построение формализованной математической модели системы; - уверенно вводит целевые функции системы, соответствующие ограничениям	- Способен реализовать построение формализованной математической модели системы; - способен вводить целевые функции системы, соответствующие ограничениям	- Не уверенно реализует построение формализованной математической модели системы; - не уверенно вводит целевые функции системы, соответствующие ограничениям	- Не способен реализовать построение формализованной математической модели системы; - не способен вводить целевые функции системы, соответствующие ограничениям
	ИПК-3.2. Адаптирует формализованную математическую модель системы (подсистемы) к изменению требований (ограничений к целевой функции) к системе (подсистеме).	ОР-ПК-3.2. Обучающийся сможет: - адаптировать формализованную математическую модель системы к изменению требований к системе	Грамотно адаптирует формализованную математическую модель системы к изменению требований к системе	Способен адаптировать формализованную математическую модель системы к изменению требований к системе	Не уверенно адаптирует формализованную математическую модель системы к изменению требований к системе	Не способен адаптировать формализованную математическую модель системы к изменению требований к системе
	ИПК-3.3. Выявляет и формализовывает в виде математической модели возникающие при функционировании системы (подсистемы) риски; выявляет и анализирует проблемные ситуации.	ОР-ПК-3.3. Обучающийся сможет: - формализовать в виде математической модели возникающие при функционировании системы риски; - выявлять и анализировать проблемные ситуации	- Уверенно формализует в виде математической модели возникающие при функционировании системы риски; - выявлять и грамотно анализирует проблемные ситуации	- Способен формализовать в виде математической модели возникающие при функционировании системы риски; - способен выявлять и анализировать проблемные ситуации	- Не уверенно формализует в виде математической модели возникающие при функционировании системы риски; - не всегда выявляет и не уверенно анализирует проблемные ситуации	- Не способен формализовать в виде математической модели возникающие при функционировании системы риски; - не способен выявлять и анализировать проблемные ситуации

2. Этапы формирования компетенций и виды оценочных средств

№	Этапы формирования компетенций (разделы практики)	Код и наименование результатов обучения	Вид оценочного средства (тесты, задания, кейсы, вопросы и др.)
1.	Организационный		
2.	Ознакомительный		
3.	Проектный	ОР-ОПК-2.1, ОР-ОПК-2.2, ОР-ОПК-2.3, ОР-ОПК-2.4, ОР-ОПК-4.1, ОР-ОПК-4.2, ОР-ОПК-4.3, ОР-ПК-2.1, ОР-ПК-2.2, ОР-ПК-2.3, ОР-ПК-3.1, ОР-ПК-3.2, ОР-ПК-3.3	Задания для проведения текущего контроля
4.	Заключительный	ОР-ОПК-2.2, ОР-ОПК-4.1, ОР-ОПК-4.2, ОР-ОПК-4.3, ОР-ОПК-4.4, ОР-ПК-2.2	Отчет по практике, Задания для зачета

3. Типовые контрольные задания или иные материалы, необходимые для оценки образовательных результатов обучения

3.1. Типовые задания для проведения текущего контроля успеваемости по практике.

Для оценки усвоения материала и приобретения навыка объектно-ориентированного программирования, студентам предлагается реализовать следующие классы.

- 1) Простой класс: дробь, круг, прямоугольник, прямоугольный треугольник, свободный вектор, прямая на плоскости, плоскость в пространстве, точка, отрезок, время, угол.
- 2) Класс «Массив».
- 3) Приложение с графическим интерфейсом «Калькулятор для массива»
- 4) Класс-шаблон «Массив».
- 5) Класс «Булев вектор произвольной длины».
- 6) Агрегированный класс «Булева матрица».
- 7) Агрегированный класс «Список»
- 8) Класс «Множество», наследник класса «Булев вектор произвольной длины»

3.2. Типовые задания для проведения промежуточной аттестации по практике

Билет №1.

Реализовать агрегированный класс – однонаправленный линейный список List.

Член-данные класса List: 1) указатель на голову(head); 2) size – количество элементов;

Методы: List (); ~List(); AddToHead(int) – добавление элемента в голову; AddToPos(int) – добавление элемента на заданную позицию; DelHead()– удаление элемента из головы; DelPos(int) – удаление элемента по позиции; void reOrder() – метод перестроения списка по принципу: сначала идут все элементы с положительными значениями, затем с отрицательными, порядок следования элементов в блоках сохранить.

Перегрузка операторов: operator- – удаление хвоста; operator== – сравнение.

Билет №2.

Реализовать агрегированный класс – матрица Matrix.

Член-данные класса Matrix: 1) Array *Line – массив строк матрицы; 2) M – количество строк матрицы; 3) N – количество столбцов матрицы.

Методы: Matrix (); Matrix (int m, int n); ~Matrix (); Print() – вывод матрицы; void ShiftTop() – циклический сдвиг строк матрицы вверх (первая строка становится последней, вторая строка - первой, третья – второй и т.д.)

Перегрузка операторов: operator =; operator+ – сложение матриц.

Класс Array – массив целых чисел можно реализовать в самом минимальном варианте (конструкторы, деструкторы, перегрузка оператора присвоения).

Билет №3.

Реализовать агрегированный класс – матрица Matrix.

Член-данные класса Matrix:

Член-данные класса Matrix: 1) Array *Line – массив строк матрицы; 2) M – количество строк матрицы; 3) N – количество столбцов матрицы.

Методы: Matrix (); Matrix (int m, int n); Matrix(const Matrix&); ~Matrix(); Scan() – ввод матрицы; int Track – след матрицы (сумма элементов главной диагонали матрицы).

Перегрузка оператора: operator* – умножение матрицы на число

Класс Array – массив целых чисел можно реализовать в самом минимальном варианте (конструкторы, деструкторы, перегрузка оператора присвоения).

Билет №4.

Реализовать класс – полином Polinom.

Член-данные класса Polinom: 1) N – степень многочлена; 2) int *A – массив коэффициентов.

Методы: Polinom (); Polinom (int *b, int n); Polinom (const Matrix&); ~ Polinom(); Scan() – ввод полинома

Перегрузка оператора: operator* – умножение полинома на число, operator-= – вычитание полиномов, потокового ввода

Билет №5.

Реализовать класс – полином Polinom.

Член-данные класса Polinom: 1) N – степень многочлена; 2) int *A – массив коэффициентов.

Методы: Polinom (); Polinom (int *b, int n); Polinom (const Matrix&); ~ Polinom(); Print() – вывод полинома

Перегрузка оператора: operator*= – умножение полинома на число, operator- – вычитание полиномов, потокового вывода

4. Методические материалы, определяющие процедуры оценивания образовательных результатов обучения

4.1. Методические материалы для оценки текущего контроля успеваемости по практике.

Задания, выполняемые студентами во время практики оцениваются следующим образом:

- оценка «отлично» выставляется, если студент реализовал все методы класса; допускается реализация отдельных методов не самым оптимальным образом;

- оценка «хорошо» выставляется, если студент реализовал 90% методов или реализованы все методы, но для большей части методов использованы не самые эффективные алгоритмы;

- оценка «удовлетворительно» выставляется, если студент реализовал не менее 70% методов класса или реализованы все методы, но для большей части реализация построена на неэффективных алгоритмах, либо не учитывает появление исключений;

- оценка «неудовлетворительно» выставляется, если студент реализовал менее 70% методов класса, либо в реализациях не менее 20% методов имеются грубые ошибки алгоритмического характера.

Для каждого задания устанавливается срок выполнения. Задания, сданные позже установленного срока без уважительной причины, оцениваются на балл ниже реальной оценки.

4.2. Методические материалы для проведения промежуточной аттестации по практике.

1) Промежуточная аттестация проводится в форме зачета на итоговом учебном занятии путем письменного ответа на билет. Билет содержит практическое задание, которое и призвано продемонстрировать полученные во время практики практические умения и навыки в области объектно-ориентированного программирования. Студент, получивший «отлично» за все текущие задания практики, получает зачет автоматически.

2) Оценка сформированности результатов обучения осуществляется руководителем практики на основе анализа предоставленных отчетных документов, выступления обучающегося и его ответов на вопросы.

Результаты прохождения практики определяются оценками «зачтено», «незачтено».

Оценка «**зачтено**» выставляется, если:

- а) студент выполнил не менее 75% заданий текущего контроля на положительную оценку;
- б) представленный студентом на зачете код верен или содержит ошибки синтаксического характера;
- в) код большинства методов оптимален, легко читаем, при написании кода использованы эффективные алгоритмы.

Оценка «**не зачтено**» выставляется, если выполнено одно из следующих условий:

- а) студент выполнил менее 75% заданий текущего контроля на положительную оценку;
- б) в представленном студенте на зачете классе реализованы не все методы;
- в) код двух и более методов класса содержит серьезные алгоритмические ошибки.