

Приложение 1  
к программе учебной (производственной) практики

**ФОНД ОЦЕНОЧНЫХ МАТЕРИАЛОВ (ОЦЕНОЧНЫХ СРЕДСТВ)**

**для проведения текущего контроля и промежуточной аттестации  
по учебной (производственной) практике**

**Б2.О.01 (У) Учебная практика технологическая**

**Автор:**

Марсенич Ирина Анатольевна,  
старший преподаватель  
кафедры точных и естественных наук СВГУ

  
подпись

19.11.2021  
дата

**И.о. зав. кафедрой точных и естественных наук**

Сироткин Андрей Вячеславович,  
кандидат технических, доцент, доцент  
кафедры ТиЕН

  
подпись

19.11.21  
дата

**Лист визирования  
Фонда оценочных материалов (оценочных средств)**

Фонд оценочных материалов (оценочных средств) по дисциплине **Б2.О.01(У) Учебная практика технологическая** проанализирован и признан актуальным для использования на 2022-2023 учебный год.

Протокол № 1 заседания кафедры *точных и естественных наук*  
от « 15 » сентября 2022 г.

И.о. заведующего кафедрой точных и естественных наук  
наименование кафедры

Крашенинникова Галина Геннадьевна, к.п.н., доцент каф.ТиЕН

ФИО заведующего кафедрой

  
подпись

15.09.22 г.  
дата

## **1. Паспорт оценочных материалов (оценочные средства)**

Оценочные материалы (оценочные средства) прилагаются к программе Б2.О.01 Учебная практика технологическая, вид практики: учебная, тип практики: технологическая, и представляют собой совокупность контрольно-измерительных материалов (индивидуальные задания, контрольные работы, тесты и др.) и методов их использования, предназначенных для измерения уровня достижения обучающимися установленных результатов обучения.

Оценочные материалы (оценочные средства) используются при проведении текущего контроля успеваемости и промежуточной аттестации обучающихся.

### **1.1. Перечень компетенций с указанием этапов их формирования в процессе прохождения практики**

Перечень компетенций, формируемых в процессе прохождения практики: (указать код и наименование компетенций)

**ОПК-7** Способен разрабатывать алгоритмы и программы, пригодные для практического применения

**ПК-2** Способен разрабатывать и адаптировать прикладное программное обеспечение

Конечным результатом прохождения практики являются сформированные когнитивные дескрипторы «знать», «уметь», «иметь практический опыт», расписанные по отдельным компетенциям. Формирование дескрипторов происходит в течение периода прохождения практики по этапам в рамках контактной работы, включающей различные виды занятий и самостоятельной работы, с применением различных форм и методов обучения (таблица 1).

Таблица 1

## Формирование компетенций в процессе прохождения практики

Код компетенции	Индикаторы достижения и уровень освоения	Дескрипторы компетенций (результаты обучения, показатели достижения результата обучения, которые обучающийся может продемонстрировать)	Контролируемые этапы прохождения практики	Оценочные материалы (оценочные средства), используемые для оценки уровня сформированности компетенций	Критерии оценивания компетенций
ОПК-7	Знать основные языки программирования и работы с базами данных, операционные системы и оболочки, современные программные среды разработки информационных систем и технологий.	<p><b>Уровень 1</b></p> <p><i>Компетенции не сформированы</i></p> <p>Знания поверхностные и фрагментарные, не способен, в общем раскрыть базовые конструкции, принципы основных языков программирования и работы с базами данных, операционными системами и оболочками, современными программными средствами разработки информационных систем и технологий.</p>	<ul style="list-style-type: none"> <li>- организационный;</li> <li>- основной (экспериментальный);</li> <li>- заключительный этап (обработка и анализ полученной информации, подготовка отчета по практике)</li> </ul>	<p>Отчет по практике Вопросы к зачету (см. п.3)</p>	<p>Полнота, системность, прочность знаний, обобщенность знаний.</p>

		ми программными средствами разработки информационных систем и технологий		
		<p><i>Уровень 3</i></p> <p><i>Повышенный</i></p> <p>Демонстрирует достаточный уровень знаний базовых конструкций, принципов основных языков программирования и работы с базами данных, операционными системами и оболочками, современными программными средствами разработки информационных систем и технологий</p>		
		<p><i>Уровень 4</i></p> <p><i>Высокий</i></p> <p>Демонстрирует глубокие знания основ базовых конструкций, принципов основных языков программирования и работы с базами данных, операционными системами и оболочками, современными программными средствами разработки информационных систем и технологий</p>		
<b>Уметь</b>	применять языки программирования и работы с базами данных, современные среды разработки информационных систем и тех-	<p><i>Уровень 1</i></p> <p><i>Компетенции не сформированы</i></p> <p>Неуверенно, с большими ошибками и недочетами умеет применять языки программирования и работы с базами данных, современные программные среды разработки информационных систем и тех-</p>	<ul style="list-style-type: none"> <li>- организационный;</li> <li>- основной (экспериментальный);</li> <li>- заключительный этап (обработка и анализ полученной информации, под-</li> </ul>	<p>Степень самостоятельности выполнения действия (умения).</p> <p>Отчет по практике Вопросы к зачету (см. п.3)</p>

нодологий для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ.	работки информационных систем и технологий для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ.	готовка отчета по практике)	
	<p><i>Уровень 2</i></p> <p><i>Пороговый уровень</i></p> <p>Неуверенно, с мелкими ошибками и недочетами умеет применять языки программирования и работы с базами данных, современные программные среды разработки информационных систем и технологий для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ.</p>		
	<p><i>Уровень 3</i></p> <p><i>Повышенный</i></p> <p>Уверенно умеет применять языки программирования и работы с базами данных, современные программы среды разработки информационных систем и технологий для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ.</p>		<p><i>Уровень 4</i></p>


		прототипов технических комплексов задач	программно-	
		<b>Уровень 4 Высокий</b>  Уверенно, профессионально применяет программирование, отладку и тестирование прототипов программно-технических комплексов задач		
ПК-2	Знать современные технологии разработки и адаптации прикладного программного обеспечения, их достоинства и недостатки	<p><b>Уровень 1</b> <i>Компетенции не сформированы</i> Знания поверхностные и фрагментарные, не способен в общем раскрыть основы современных технологий разработки и адаптации прикладного программного обеспечения, их достоинства и недостатки</p> <p><b>Уровень 2</b> <i>Прогрессивный уровень</i> Знания поверхностные и фрагментарные, способен в общем раскрыть основы современных технологий разработки и адаптации прикладного программного обеспечения, их достоинства и недостатки</p> <p><b>Уровень 3</b> <i>Повышенный</i> Демонстрирует достаточный уровень знаний основ современных технологий разработки и</p>	<ul style="list-style-type: none"> <li>- организационный;</li> <li>- основной (экспериментальный);</li> <li>- заключительный</li> </ul> <p>этап (обработка и анализ полученной информации, подготовка отчета по практике)</p>	<p>Отчет по практике Вопросы к зачету (см. п.3)</p> <p>Полнота, системность, прочность знаний, обобщенность знаний.</p>

Ф СВГУ «ФОС ПРАКТИКА ФГОС 3++»


		<i>Высокий</i>		
	Уверенно, профессионально умеет разрабатывать, адаптировать компоненты прикладного программного обеспечения			
<b>Иметь практический опыт разработки прикладного программного обеспечения на современных языках программирования, методами адаптации прикладного программного обеспечения</b>	<p><i>Уровень 1</i></p> <p><i>Компетенции не сформированы</i></p> <p>Неуверенно, с большими ошибками и недочетами применяет способы разработки прикладного программного обеспечения на современных языках программирования, методами адаптации прикладного программного обеспечения</p> <p><i>Уровень 2</i></p> <p><i>Пороговый уровень</i></p> <p>Неуверенно, с мелкими ошибками и недочетами применяет способы разработки прикладного программного обеспечения на современных языках программирования, методами адаптации прикладного программного обеспечения</p> <p><i>Уровень 3</i></p> <p><i>Повышенный</i></p> <p>Уверенно применяет способы разработки прикладного программного обеспечения на современных языках программирования, методами адаптации при-</p>	<ul style="list-style-type: none"> <li>- организационный;</li> <li>- основной (экспериментальный);</li> <li>- заключительный этап (обработка и анализ полученной информации, подготовка отчета по практике)</li> </ul>	<p>Отчет по практике Вопросы к зачету (см. п.3)</p>	Степень самостоятельности го применения.

		прикладного программного обеспечения
	<i>Уровень 4 Высокий</i>	Уверенно, профессионально применяет способы разработки прикладного программного обеспечения на современных языках программирования, методами адаптации прикладного программного обеспечения

## **1.2. Сведения о дисциплинах (модулях), участвующих в формировании данных компетенций**

Б2.О.01 Учебная практика технологическая, вид практики: учебная, тип практики: технологическая – относится к Блоку 2 «Практики» учебного плана ОПОП ВО по направлению подготовки 09.03.03 «Прикладная информатика», профиля подготовки «Прикладная информатика и информационная безопасность». Программа базируется на компетенциях, сформированных у обучающихся в ходе изучения дисциплин Блока 1 «Дисциплины (модули)» учебного плана ОПОП ВО:

Б1.О.12 ПО ЭВМ. Офисные приложения, утилиты; Б1.О.10 Математическая логика и теория алгоритмов; Б1.О.14 Программирование; Б1.В.04 Разработка и администрирование WEB-приложений; Б1.В.05 Методы оценки безопасности компьютерных систем; Б1.В.06 Объектно-ориентированное программирование; Б1.В.08 Системное программирование; Б1.В.10 Программирование на языках высокого уровня; Б1.В.18 Программная инженерия; Б1.В.19 Прикладное программирование.

## **1.3. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания**

Оценивание результатов прохождения практики осуществляется в соответствии с Положением о текущем контроле успеваемости и промежуточной аттестации студентов СВГУ.

Текущий контроль представляет собой проверку усвоения учебного материала теоретического и практического характера, регулярно осуществляющуюся в процессе прохождения обучающимся практики. Текущий контроль успеваемости осуществляется в течение периода прохождения практики по установленным формам организации практики и самостоятельной работы студента в соответствии с утвержденным в установленном порядке календарным учебным графиком.

К формам текущей контроля по программе практики относятся: устный опрос, письменные задания, самостоятельная работа с литературой, консультации преподавателей и руководителей от предприятия, индивидуальные задания, анализ конкретных ситуаций, собеседование и т.п. (список м.б. дополнен).

Текущий контроль по практике проводится с целью обеспечения своевременной обратной связи, для коррекции заданий по практике, активизации самостоятельной работы обучающихся и помогает оценить более крупные совокупности знаний и умений, в некоторых случаях – даже формирование определенных профессиональных компетенций.

К достоинствам данного типа относится его систематичность, непосредственно коррелирующая с требованием постоянного и непрерывного мониторинга качества обучения, а также возможность оценки успеваемости обучающегося. При текущем контроле успеваемости акцент делается на установлении подробной, реальной картины достижений и успешности усвоения обучающимся учебной программы на данный момент времени.

Недостатками текущего контроля являются фрагментарность и локальность проверки. Компетенцию целиком, а не отдельные ее элементы (знания, умения, практический опыт) при подобном контроле проверить невозможно.

Текущий контроль и промежуточная аттестация служат основным средством обеспечения в учебном процессе «обратной связи» между преподавателем и обучающимся, необходимой для стимулирования работы обучающихся и совершенствования методики преподавания дисциплин (модулей).

Оценивание знаний, умений, практического опыта должно носить комплексный, системный характер – с учетом как места дисциплины (модуля) в структуре ОПОП ВО, так и содержательных и смысловых внутренних связей. Связи формируемых компетенций с модулями, разделами (темами) дисциплины обеспечивают возможность реализации для

текущего контроля, промежуточной аттестации по дисциплине и итогового контроля наиболее подходящих оценочных средств.

Промежуточная аттестация по практике Б2.О.01 Учебная практика технологическая проводится в форме зачета с оценкой.

В таблице 2 приведена оценка уровня сформированности компетенций на основании критериев модульно-рейтинговой системы.

Таблица 2

<b>№ уровня</b>	<b>Уровни сформированности компетенции</b>	<b>Основные признаки уровня</b>
<b>1</b>	<b>Компетенции не сформированы</b>	<b>менее 50%</b>
<b>2</b>	<b>Пороговый уровень</b>	<b>50-65%</b>
<b>3</b>	<b>Повышенный уровень</b>	<b>65-85%</b>
<b>4</b>	<b>Высокий уровень</b>	<b>85-100%</b>

Структура оценочных материалов (оценочных средств), позволяющих оценить уровень компетенций обучающихся при прохождении практики Б2.О.01 Учебная практика технологическая, вид практики: производственная, тип практики: преддипломная

Код компетенции	Знать	Оценочное средство		Оценочное средство		Оценочное средство
		Текущий контроль	Промежуточный контроль	Текущий контроль	Промежуточный контроль	
ОПК-7	Знать основные языки и программыирования и работы с базами данных, операционные системы и оболочки, современные программы разработки информационных систем и технологий.	Отчет по практике Вопросы к зачету (см. п.3)	Уметь применять языки программирования и работы с базами данных, современные программы разработки информационных систем и технологий для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ	–	Отчет по практике Вопросы к зачету (см. п.3)	Иметь практический опыт программирования, отладки и тестирования прототипов программно-технических комплексов задач
ПК-2	Знать современные технологии разработки и адаптации прикладного программного обеспечения, их достоинства и недостатки	Отчет по практике Вопросы к зачету (см. п.3)	Уметь разрабатывать, адаптировать компоненты прикладного программного обеспечения	–	Отчет по практике Вопросы к зачету (см. п.3)	Иметь практический опыт разработки прикладного программного обеспечения на современных языках программирования, методами адаптации прикладного программного обеспечения

Примечание: \* из программы практики; \*\* подготовка презентации и доклада, отчета и т.п.

Экзамен, зачет, зачет с оценкой является формой итоговой оценки качества освоения обучающимся образовательной программы по практике в целом. По результатам экзамена, зачета с оценкой обучающемуся выставляется оценка «отлично», «хорошо», «удовлетворительно», или «неудовлетворительно».

Зачет, оценка «отлично» за зачет с оценкой или экзамен (свыше 85 баллов) выставляется обучающемуся, если:

- обучающийся своевременно выполнил весь объем работы, требуемый программой практики, показал глубокую теоретическую, методическую, профессионально-прикладную подготовку;

- обучающийся умело применил полученные знания во время прохождения практики, показал владение традиционными и альтернативными методами, современными приемами в рамках своей профессиональной деятельности, точно использовал профессиональную терминологию;

- обучающийся ответственно и с интересом относился к своей работе, грамотно, в соответствии с требованиями сделал анализ проведенной работы;

- обучающийся выполнил отчет по практике в полном объеме, результативность практики представлена в количественной и качественной обработке, продуктах деятельности;

- обучающийся демонстрирует повышенный уровень сформированности необходимых компетенций.

Компетенция(и) или ее часть(и) сформированы на высоком уровне (уровень 4) (см. таблицу 2).

Зачет, оценка «хорошо» за зачет с оценкой или экзамен (66-85 баллов) выставляется обучающемуся, если:

- обучающийся демонстрирует достаточно полные знания всех профессионально-прикладных и теоретических вопросов в объеме программы практики;

- обучающийся полностью выполнил программу, но допустил незначительные ошибки при выполнении задания, владеет инструментарием методики в рамках своей профессиональной подготовки, умением использовать его;

- обучающийся грамотно использует профессиональную терминологию при оформлении отчетной документации по практике;

- обучающийся демонстрирует повышенный уровень сформированности необходимых компетенций.

Компетенция(и) или ее часть(и) сформированы на повышенном уровне (уровень 3) (см. таблицу 2).

Зачет, оценка «удовлетворительно» за зачет с оценкой или экзамен (50-65 баллов) выставляется обучающемуся, если:

- обучающийся выполнил программу практики, однако в процессе работы не проявил достаточной самостоятельности, инициативы и заинтересованности, допустил существенные ошибки при выполнении заданий практики;

- обучающийся демонстрирует недостаточный объем знаний и низкий уровень их применения на практике, неосознанное владение инструментарием;

- обучающийся показал низкий уровень владения методической терминологией, низкий уровень владения профессиональным стилем речи, низкий уровень оформления документации по практике;

- обучающийся демонстрирует пороговый уровень сформированности необходимых компетенций.

Компетенция(и) или ее часть(и) сформированы на пороговом уровне (уровень 2) (см. таблицу 2).

Не зачет, оценка «неудовлетворительно» за зачет с оценкой или экзамен (менее 50 балла) выставляется обучающемуся, если:

- обучающийся владеет фрагментарными знаниями и не умеет применить их на практике;
- обучающийся не выполнил программу практики, не получил положительной характеристики, не проявил инициативу, не представил рабочие материалы, не проявил склонностей и желания к работе, не представил необходимую отчетную документацию;
- обучающийся не явился на практику;
- обучающийся демонстрирует уровень сформированности необходимых компетенций ниже порогового.

Компетенция(и) или ее часть(и) не сформированы (уровень 1) (см. таблицу 2).

## **2. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, практического опыта, характеризующие этапы формирования компетенций в процессе прохождения практики**

### **2.1. Формы текущего контроля**

Не предусмотрены.

### **2.2. Формы промежуточного контроля**

Форма промежуточного контроля:

на дневном отделении:

- 1 курс (2 семестр) – зачет с оценкой,
- 2 курс (4 семестр) – зачет с оценкой,
- 3 курс (6 семестр) – зачет с оценкой,

на заочном отделении:

- 1 курс – зачет с оценкой,
- 2 курс – зачет с оценкой,
- 3 курс – зачет с оценкой.

## **3. Методические материалы, определяющие процедуры оценивания знаний, умений, практического опыта**

По видам индивидуальных заданий приводится описание того или обсуждается устно, каким образом необходимо выполнить данное индивидуальное задание, способы и механизмы его выполнения, выбор номера варианта и др.

Методические материалы, определяющие процедуру оценивания знаний, умений, практического опыта, характеризующих этапы формирования компетенций:

1 Промежуточный контроль.

1.1 Наличие отчета по практике.

1.2 Вопросы к зачету.

### **А) Примерные индивидуальные задания на практику**

Индивидуальное задание на практикудается в соответствии с профилем подготовки – руководителем практики от университета.

В качестве индивидуальных заданий студенты разрабатывают Windows-приложение, создают меню с разными командами, создают параметризованную коллекцию для хранения описанной структуры, пишут Windows-приложение, которое выполняет анимацию изображения.

Примерные варианты индивидуальных заданий:

### **Задание 1. Диалоговые окна**

**Общая часть задания:** написать Windows-приложение, заголовок главного окна которого содержит Ф. И. О., группу и номер варианта. В программе должна быть предусмотрена обработка исключений, возникающих из-за ошибочного ввода пользователя.

#### **Вариант 1**

Создать меню с командами Input, Calc и Exit.

При выборе команды Input открывается диалоговое окно, содержащее:

- три поля типа TextBox для ввода длин трех сторон треугольника;
- группу из двух флажков (Периметр и Площадь) типа CheckBox;
- кнопку типа Button.

Обеспечить возможность:

- ввода длин трех сторон треугольника;
- выбора режима с помощью флажков: подсчет периметра и/или площади треугольника.

При выборе команды Calc открывается диалоговое окно с результатами. При выборе команды Exit приложение завершается.

#### **Вариант 2**

Создать меню с командами Size, Color, Paint, Quit.

Команда Paint недоступна. При выборе команды Quit приложение завершается.

При выборе команды Size открывается диалоговое окно, содержащее:

- два поля типа TextBox для ввода длин сторон прямоугольника;
- группу из трех флажков (Red, Green, Blue) типа CheckBox;
- кнопку типа Button.

Обеспечить возможность:

- ввода длин сторон прямоугольника в пикселях в поля ввода;
- выбора его цвета с помощью флажков.

После задания параметров команда Paint становится доступной.

При выборе команды Paint в главном окне приложения выводится прямоугольник заданного размера и сочетания цветов или выдается сообщение, если введенные размеры превышают размер окна.

#### **Вариант 3**

Создать меню с командами Input, Work, Exit.

При выборе команды Exit приложение завершает работу. При выборе команды Input открывается диалоговое окно, содержащее:

- три поля ввода типа TextBox с метками Radius, Height, Density;
- группу из двух флажков (Volume, Mass) типа CheckBox;
- кнопку типа Button.

Обеспечить возможность:

- ввода радиуса, высоты и плотности конуса;
- выбора режима с помощью флажков: подсчет объема и/или массы конуса.

При выборе команды Work открывается окно сообщений с результатами.

#### **Вариант 4**

Создать меню с командами Input, Calc, Draw, Exit.

При выборе команды Exit приложение завершает работу. При выборе команды Input открывается диалоговое окно, содержащее:

- поле ввода типа TextBox с меткой Radius;
- группу из двух флажков (Square, Length) типа CheckBox;
- кнопку типа Button.

Обеспечить возможность:

- ввода радиуса окружности;
- выбора режима с помощью флажков: подсчет площади круга (Square) и/или длины окружности (Length).

При выборе команды Calc открывается окно сообщений с результатами. При выборе команды Draw в центре главного окна выводится круг введенного радиуса или выдается сообщение, что рисование невозможно (если диаметр превышает размеры рабочей области).

#### **Вариант 5**

Создать меню с командами input, Calc, About.

При выборе команды About открывается окно с информацией о разработчике.

При выборе команды Input открывается диалоговое окно, содержащее:

- три поля ввода типа TextBox с метками Number 1, Number 2, Number 3;
- группу из двух флажков (Summ, Least multiple) типа CheckBox;
- кнопку типа Button.

Обеспечить возможность ввода трех чисел и выбора режима вычислений с помощью флажков: подсчет суммы трех чисел (Summ) и/или наименьшего общего кратного двух первых чисел (Least multiple). При выборе команды Calc открывается диалоговое окно с результатами.

#### **Вариант 6**

Создать меню с командами Input, Calc, Quit.

Команда Calc недоступна. При выборе команды Quit приложение завершается.

При выборе команды Input открывается диалоговое окно, содержащее:

- два поля ввода типа TextBox с метками Number 1, Number 2;
- группу из трех флажков (Summa, Max divisor, Multiply) типа CheckBox;
- кнопку типа Button.

Обеспечить возможность:

- ввода двух чисел;
- выбора режима вычислений с помощью флажков (можно вычислять в любой комбинации такие величины, как сумма, наибольший общий делитель и произведение двух чисел).

При выборе команды Calc открывается окно сообщений с результатами.

#### **Вариант 7**

Создать меню с командами Begin, Help, About.

При выборе команды About открывается окно с информацией о разработчике.

При выборе команды Begin открывается диалоговое окно, содержащее:

- поле ввода типа TextBox с меткой input;
- метку типа Label для вывода результата;
- группу из трех переключателей (2, 8, 16) типа RadioButton;
- две кнопки типа Button — Do и OK.

Обеспечить возможность:

- ввода числа в десятичной системе в поле input;
- выбора режима преобразования с помощью переключателей: перевод в двоичную, восьмеричную или шестнадцатеричную систему счисления.

При щелчке на кнопке Do должен появляться результат перевода.

#### **Вариант 8**

Создать меню с командами Input color, Change, Exit, Help.

При выборе команды Exit приложение завершает работу. При выборе команды Input color открывается диалоговое окно, содержащее:

- три поля ввода типа TextBox с метками Red, Green, Blue;
- группу из двух флажков (Left, Right) типа CheckBox;

- кнопку типа Button.

Обеспечить возможность ввода RGB-составляющих цвета. При выборе команды Change цвет главного окна изменяется на заданный (левая, правая или обе половины окна в зависимости от установки флажков).

#### **Вариант 9**

Создать меню с командами Input size, Choose, Change, Exit.

При выборе команды Exit приложение завершает работу. Команда Change недоступна.

При выборе команды Input size открывается диалоговое окно, содержащее:

- два поля ввода типа TextBox с метками Size x, Size y;
- кнопку типа Button.

При выборе команды Choose открывается диалоговое окно, содержащее:

- группу из двух переключателей (Increase, Decrease) типа RadioButton;
- кнопку типа Button.

Обеспечить возможность ввода значений в поля Size x и Size y. Значения интерпретируются как количество пикселов, на которое надо изменить размеры главного окна (увеличить или уменьшить в зависимости от положения переключателей).

После ввода значений команда Change становится доступной. При выборе этой команды размеры главного окна увеличиваются или уменьшаются на введенное количество пикселов.

#### **Вариант 10**

Создать меню с командами Begin, Work, About.

При выборе команды About открывается окно с информацией о разработчике.

При выборе команды Begin открывается диалоговое окно, содержащее:

- поле ввода типа TextBox с меткой Input word;
- группу из двух переключателей (Upper case, Lower case) типа RadioButton;
- кнопку типа Button.

Обеспечить возможность ввода слова и выбора режима перевода в верхний или нижний регистр в зависимости от положения переключателей. При выборе команды Work открывается диалоговое окно с результатом перевода.

#### **Вариант 11**

Создать меню с командами Input color, Change, Clear.

При выборе команды Input color открывается диалоговое окно, содержащее:

- группу из двух флажков (Up, Down) типа CheckBox;
- группу из трех переключателей (Red, Green, Blue) типа RadioButton;
- кнопку типа Button.

Обеспечить возможность:

- выбора цвета с помощью переключателей;
- ввода режима, определяющего, какая область закрашивается: все окно, его верхняя или нижняя половина.

При выборе команды Change цвет главного окна изменяется на заданный (верхняя, нижняя или обе половины в зависимости от введенного режима). При выборе команды Clear восстанавливается первоначальный цвет окна.

#### **Вариант 12**

Создать меню с командами Translate, Help, About, Exit.

При выборе команды Exit приложение завершает работу. При выборе команды

Translate открывается диалоговое окно, содержащее:

- поле ввода типа TextBox с меткой Binary number;
- поле ввода типа TextBox для вывода результата (read-only);
- группу из трех переключателей (8, 10, 16) типа RadioButton;

- кнопку Do типа Button.

Обеспечить возможность:

- ввода числа в двоичной системе в поле Binary number;
- выбора режима преобразования с помощью переключателей: перевод в восьмеричную, десятичную или шестнадцатеричную систему счисления.

При щелчке на кнопке Do должен появляться результат перевода.

### Вариант 13

Создать меню с командами Reverse, About, Exit.

При выборе команды About открывается окно с информацией о разработчике.

При выборе команды Reverse открывается диалоговое окно, содержащее:

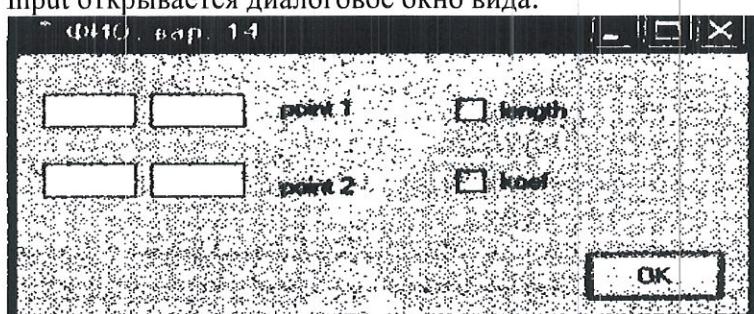
- поле ввода типа TextBox с меткой Input;
- группу из двух переключателей (Upper case, Reverse) типа CheckBox;
- кнопку OK типа Button.

Обеспечить возможность ввода фразы и выбора режима: перевод в верхний регистр и/или изменение порядка следования символов на обратный в зависимости от состояния переключателей. Результат преобразования выводится в исходное поле ввода.

### Вариант 14

Создать меню с командами Input, Show и Exit.

При выборе команды Exit приложение завершает работу. При выборе команды Input открывается диалоговое окно вида:



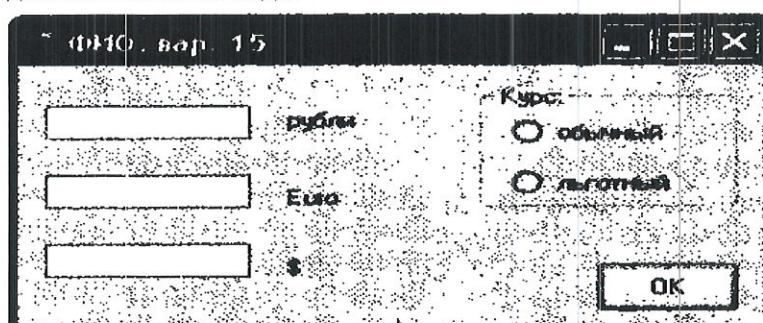
Обеспечивается возможность ввода координат двух точек и выбора режима с помощью флажков length и коef: подсчет длины отрезка, соединяющего эти точки, и/или углового коэффициента.

При выборе команды Show открывается окно сообщений с результатами подсчета.

### Вариант 15

Создать меню с командами Input, About и Exit.

При выборе команды Exit приложение завершает работу. При выборе команды About открывается окно с информацией о разработчике. При выборе команды Input открывается диалоговое окно вида:



Обеспечивается возможность ввода суммы в рублях и перевода ее в евро и доллары по-обычному или льготному курсу. Поля Euro и \$ доступны только для чтения.

### **Вариант 16**

Создать меню с командами Begin, Work, About.

При выборе команды About открывается окно с информацией о разработчике.

При выборе команды Begin открывается диалоговое окно, содержащее:

- два поля ввода типа TextBox;
- группу из двух переключателей (First letter, All letters) типа RadioButton;
- кнопку типа Button.

Обеспечить возможность ввода предложения и выбора режима его преобразования: либо начинать с прописной буквы каждое слово (First letter), либо перевести все буквы в верхний регистр (All letters). При выборе команды Work открывается диалоговое окно с результатом преобразования.

### **Вариант 17**

Написать анализатор текстовых файлов, выводящий информацию о количестве слов в тексте, а также статистическую информацию о введенной пользователем букве.

Создать следующую систему меню:

- Файл
  - Загрузить текст
  - Выход
- Анализ
  - Количество слов
  - Повторяемость буквы

При выборе файла для загрузки использовать объект типа OpenFileDialog. При выборе команды Количество слов программа должна вывести в окно сообщений количество слов в тексте.

При выборе команды Повторяемость буквы программа предлагает пользователю ввести букву, а затем выводит количество ее повторений без учета регистра в окно сообщений.

### **Вариант 18**

Создать редактор текстовых файлов с возможностью сохранения текста в формате HTML.

Создать следующую систему меню:

- Файл
  - Загрузить текст
  - Сохранить как текст
  - Сохранить как HTML
- Выход

При выборе файла для загрузки использовать объект OpenFileDialog. При выборе файла для сохранения использовать объект SaveFileDialog. Для редактирования текста использовать объект Memo.

При сохранении текста в формате HTML текст записывать в файл с заменой:

- всех пробелов на символы  ;
- всех символов перевода строки на символы <BR>;
- всех символов < на символы &lt;;
- всех символов > На символы &gt;;
- всех символов & на символы &amp;;
- всех символов " (двойные кавычки) на символы &quot;;

### **Вариант 19**

Создать меню с командами Input, Draw, Clear.

При выборе команды Input открывается диалоговое окно, содержащее:

- четыре поля для ввода координат двух точек;
- группу из трех переключателей (Red, Green, Blue) типа RadioButton;
- кнопку типа Button.

При выборе команды Draw в главное окно выводится отрезок прямой выбранного цвета с координатами концов отрезка, заданными в диалоговом окне. При выборе команды Clear отрезок стирается.

### **Вариант 20**

Создать меню с командами Input, Change, Exit.

При выборе команды Exit приложение завершает работу. Команда Change недоступна. В центре главного окна выведен квадрат размером 100 x 100 пикселов.

При выборе команды Input открывается диалоговое окно, содержащее:

- два поля ввода типа TextBox с метками Size x, Size y;
- группу из двух переключателей (Increase, Decrease) типа RadioButton;
- кнопку типа Button.

Обеспечить возможность ввода значений в поля Size x и Size y. Значения интерпретируются как количество пикселов, на которое надо изменить размеры квадрата, выведенного в главное окно (увеличить или уменьшить в зависимости от положения переключателей).

После ввода значений команда Change становится доступной. При выборе этой команды размеры квадрата увеличиваются или уменьшаются на введенное количество пикселов. Если квадрат выходит за пределы рабочей области окна, выдается сообщение.

### **Вариант 21**

Написать Windows-приложение, которое по заданным в файле исходным данным выводит информацию о компьютерах.

Создать меню с командами Choose, Show, Quit.

Команда S h o w недоступна. Команда Q uit завершает работу приложения.

При запуске приложения из файла читаются исходные данные. Файл необходимо сформировать самостоятельно. Каждая строка файла содержит тип компьютера, цену (price) и емкость жесткого диска (hard drive).

При выборе команды Choose открывается диалоговое окно, содержащее:

- поле типа TextBox для ввода минимальной емкости диска;
- поле типа TextBox для ввода максимальной приемлемой цены;
- группу из двух переключателей (Hard drive, Price) типа RadioButton;
- OK, Cancel — кнопки типа Button.

После ввода всех данных команда Show становится доступной. Команда Show открывает диалоговое окно, содержащее список компьютеров, удовлетворяющий введенным ограничениям и упорядоченный по отмеченной характеристике.

## **Задание 2. Структуры и параметризованные коллекции**

**Общая часть задания:** Описать структуру, соответствующую заданиям, представленным в вариантах 1-20. Например, для варианта 1 – пункты 1,2,4 (сортировка по возрастанию номера группы), 6 (вывод списка всех студентов, средний балл которых больше 4,0), для остальных вариантов по аналогии.

Создать параметризованную коллекцию (см. раздел «Классы-прототипы») для хранения описанной структуры. Вид коллекции выбрать самостоятельно. Написать Windows-приложение для работы с этой коллекцией, позволяющее выполнять:

- Добавление элемента в коллекцию с клавиатуры;
- Считывание данных из файла;
- Запись данных в тот же или указанный файл;
- Сортировку данных по различным критериям;
- Поиск элемента по заданному полю;
- Вывод всех элементов, удовлетворяющих заданному условию;

■ Удаление элемента из коллекции

Приложение должно содержать меню и диалоговые окна и предусматривать обработку возможных ошибок пользователя с помощью исключений.

**Вариант 1**

Описать структуру с именем STUDENT, содержащую следующие поля:

- фамилия и инициалы;
- номер группы;
- успеваемость (массив из пяти элементов).

Написать программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из десяти структур типа STUDENT (записи должны быть упорядочены по возрастанию номера группы);
- вывод на экран фамилий и номеров групп для всех студентов, включенных в массив, если средний балл студента больше 4,0 (если таких студентов нет, вывести соответствующее сообщение).

**Вариант 2**

Описать структуру с именем STUDENT, содержащую следующие поля:

- фамилия и инициалы;
- номер группы;
- успеваемость (массив из пяти элементов).

Написать программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из десяти структур типа STUDENT (записи должны быть упорядочены по возрастанию среднего балла);
- вывод на экран фамилий и номеров групп для всех студентов, имеющих оценки 4 и 5 (если таких студентов нет, вывести соответствующее сообщение).

**Вариант 3**

Описать структуру с именем STUDENT, содержащую следующие поля:

- фамилия и инициалы;
- номер группы;
- успеваемость (массив из пяти элементов).

Написать программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из десяти структур типа STUDENT (записи должны быть упорядочены по алфавиту);
- вывод на экран фамилий и номеров групп для всех студентов, имеющих хотя бы одну оценку 2 (если таких студентов нет, вывести соответствующее сообщение).

**Вариант 4**

Описать структуру с именем AEROFL0T, содержащую следующие поля:

- название пункта назначения рейса;
- номер рейса;
- тип самолета.

Написать программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из семи элементов типа AEROFL0T (записи должны быть упорядочены по возрастанию номера рейса);
- вывод на экран номеров рейсов и типов самолетов, вылетающих в пункт назначения, название которого совпало с названием, введенным с клавиатуры (если таких рейсов нет, вывести соответствующее сообщение).

**Вариант 5**

Описать структуру с именем AEROFL0T, содержащую следующие поля:

- название пункта назначения рейса;
- номер рейса;

- тип самолета.

Написать программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из семи элементов типа AEROFL0T (записи должны быть размещены в алфавитном порядке по названиям пунктов назначения);
- вывод на экран пунктов назначения и номеров рейсов, обслуживаемых самолетом, тип которого введен с клавиатуры (если таких рейсов нет, вывести соответствующее сообщение).

#### Вариант 6

Описать структуру с именем WORKER, содержащую следующие поля:

- фамилия и инициалы работника;
- название занимаемой должности;
- год поступления на работу.

Написать программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из десяти структур типа WORKER (записи должны быть упорядочены по алфавиту);
- вывод на экран фамилий работников, стаж работы которых превышает значение, введенное с клавиатуры (если таких работников нет, вывести соответствующее сообщение).

#### Вариант 7

Описать структуру с именем TRAIN, содержащую следующие поля:

- название пункта назначения;
- номер поезда;
- время отправления.

Написать программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из восьми элементов типа TRAIN (записи должны быть размещены в алфавитном порядке по названиям пунктов назначения);
- вывод на экран информации о поездах, отправляющихся после введенного с клавиатуры времени (если таких поездов нет, вывести соответствующее сообщение).

#### Вариант 8

Описать структуру с именем TRAIN, содержащую следующие поля:

- название пункта назначения;
- номер поезда;
- время отправления.

Написать программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из шести элементов типа TRAIN (записи должны быть упорядочены по времени отправления поезда);
- вывод на экран информации о поездах, направляющихся в пункт, название которого введено с клавиатуры (если таких поездов нет, вывести соответствующее сообщение).

#### Вариант 9

Описать структуру с именем TRAIN, содержащую следующие поля:

- название пункта назначения;
- номер поезда;
- время отправления.

Написать программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из восьми элементов типа TRAIN (записи должны быть упорядочены по номерам поездов);

- вывод на экран информации о поезде, номер которого введен с клавиатуры (если таких поездов нет, вывести соответствующее сообщение).

#### **Вариант 10**

Описать структуру с именем MARSH, содержащую следующие поля:

- название начального пункта маршрута;
- название конечного пункта маршрута;
- номер маршрута.

Написать программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из восьми элементов типа MARSH (записи должны быть упорядочены по номерам маршрутов);
- вывод на экран информации о маршруте, номер которого введен с клавиатуры (если таких маршрутов нет, вывести соответствующее сообщение).

#### **Вариант 11**

Описать структуру с именем MARSH, содержащую следующие поля:

- название начального пункта маршрута;
- название конечного пункта маршрута;
- номер маршрута.

Написать программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из восьми элементов типа MARSH (записи должны быть упорядочены по номерам маршрутов);
- вывод на экран информации о маршрутах, которые начинаются или оканчиваются в пункте, название которого введено с клавиатуры (если таких маршрутов нет, вывести соответствующее сообщение).

#### **Вариант 12**

Описать структуру с именем NOTE, содержащую следующие поля:

- фамилия, имя;
- номер телефона;
- дата рождения (массив из трех чисел).

Написать программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из восьми элементов типа NOTE (записи должны быть упорядочены по дате рождения);
- вывод на экран информации о человеке, номер телефона которого введен с клавиатуры (если такого нет, вывести соответствующее сообщение).

#### **Вариант 13**

Описать структуру с именем NOTE, содержащую следующие поля:

- фамилия, имя;
- номер телефона;
- дата рождения (массив из трех чисел).

Написать программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из восьми элементов типа NOTE (записи должны быть размещены по алфавиту);
- вывод на экран информации о людях, чьи дни рождения приходятся на месяц, значение которого введено с клавиатуры (если таких нет, вывести соответствующее сообщение).

#### **Вариант 14**

Описать структуру с именем NOTE, содержащую следующие поля:

- фамилия, имя;
- номер телефона;
- дата рождения (массив из трех чисел).

Написать программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из восьми элементов типа NOTE (записи должны быть упорядочены по трем первым цифрам номера телефона);
- вывод на экран информации о человеке, чья фамилия введена с клавиатуры (если такого нет, вывести соответствующее сообщение).

### **Вариант 15**

Описать структуру с именем ZNAK, содержащую следующие поля:

- фамилия, имя;
- знак Зодиака;
- дата рождения (массив из трех чисел).

Написать программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из восьми элементов типа ZNAK (записи должны быть упорядочены по дате рождения);
- вывод на экран информации о человеке, чья фамилия введена с клавиатуры (если такого нет, вывести соответствующее сообщение).

### **Вариант 16**

Описать структуру с именем ZNAK, содержащую следующие поля:

- фамилия, имя;
- знак Зодиака;
- дата рождения (массив из трех чисел).

Написать программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из восьми элементов типа ZNAK (записи должны быть упорядочены по дате рождения);
- вывод на экран информации о людях, родившихся под знаком, название которого введено с клавиатуры (если таких нет, вывести соответствующее сообщение).

### **Вариант 17**

Описать структуру с именем ZNAK, содержащую следующие поля:

- фамилия, имя;
- знак Зодиака;
- дата рождения (массив из трех чисел).

Написать программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из восьми элементов типа ZNAK (записи должны быть упорядочены по знакам Зодиака);
- вывод на экран информации о людях, родившихся в месяц, значение которого введено с клавиатуры (если таких нет, вывести соответствующее сообщение).

### **Вариант 18**

Описать структуру с именем PRICE, содержащую следующие поля:

- название товара;
- название магазина, в котором продается товар;
- стоимость товара в рублях.

Написать программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из восьми элементов типа PRICE (записи должны быть упорядочены в алфавитном порядке по названиям товаров);
- вывод на экран информации о товаре, название которого введено с клавиатуры (если таких товаров нет, вывести соответствующее сообщение).

### **Вариант 19**

Описать структуру с именем PRICE, содержащую следующие поля:

- название товара;
- название магазина, в котором продается товар;

- стоимость товара в рублях.

Написать программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из восьми элементов типа PRICE (записи должны быть упорядочены в алфавитном порядке по названиям магазинов);
- вывод на экран информации о товарах, продающихся в магазине, название которого введено с клавиатуры (если такого магазина нет, вывести соответствующее сообщение).

### **Вариант 20**

Описать структуру с именем ORDER, содержащую следующие поля:

- расчетный счет плательщика;
- расчетный счет получателя;
- перечисляемая сумма в рублях.

Написать программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из восьми элементов типа ORDER (записи должны быть размещены в алфавитном порядке по расчетным счетам плательщиков);
- вывод на экран информации о сумме, снятой с расчетного счета плательщика, введенного с клавиатуры (если такого расчетного счета нет, вывести соответствующее сообщение).

## **Задание 3. Графика в Windows**

### **Вариант 1**

Написать Windows-приложение, которое выполняет анимацию изображения.

Создать меню с командами Show picture, Choose, Animate, Stop, Quit.

Команда Quit завершает работу приложения. При выборе команды Show picture в центре экрана рисуется объект, состоящий из нескольких графических примитивов.

При выборе команды Choose открывается диалоговое окно, содержащее:

- поле типа TextBox с меткой Speed для ввода скорости движения объекта;
- группу Direction из двух переключателей (Up-Down, Left-Right) типа RadioButton для выбора направления движения;
- кнопку типа Button.

По команде Animate объект начинает перемещаться в выбранном направлении до края окна и обратно с заданной скоростью, по команде Stop – прекращается движение.

### **Вариант 2**

Написать Windows-приложение, которое по заданным в файле исходным строит график или столбиковую диаграмму.

Создать меню с командами Input data, Choose, Line, Bar, Quit.

Команды Line и Bar недоступны. Команда Quit завершает работу приложения.

При выборе команды Input data из файла читаются исходные данные (файл сформировать самостоятельно).

По команде Choose открывается диалоговое окно, содержащее:

- список для выбора цвета графика типа TListBox;
- группу из двух переключателей (Line, Bar) типа RadioButton;
- кнопку типа Button

Обеспечить возможность ввода цвета и выбора режима: построение графика (Line) или столбиковой диаграммы (Bar). После указания параметров становится доступной соответствующая команда меню.

По команде Line или Bar в заглавном окне приложения выбранным цветом строится график и диаграмма. Окно должно содержать заголовок графика или диаграммы, наименова-

ние и градацию осей. Изображение должно занимать все окно и масштабироваться при изменении размеров окна.

### Вариант 3

Написать Windows-приложение, которое строит графики четырех заданных функций.

Создать меню с командами Chart, Build, Clear, About, Quit.

Команда Quit завершает работу приложения. При выборе команды About открывается окно с информацией о разработчике.

Команда Chart открывает диалоговое окно, содержащее:

- список для выбора цвета графики типа TListBox;
- список для выбора типа графика типа TListBox, содержащей четыре пункта:  
 $\sin(x)$ ,  $\sin(x+\pi/4)$ ,  $\cos(x)$ ,  $\cos(x-\pi/4)$ ;
- кнопку типа Button

Обеспечить возможность выбора цвета и вида графика. После щелчка на кнопке ОК в главном окне приложения строится график выбранной функции на интервале от  $-\pi/2$  до  $+\pi/2$ . Окно должно содержать заголовок графика, наименование и градацию осей. Изображение должно занимать все окно и масштабироваться при изменении размеров окна.

Команда Clear очищает окно.

### Б) Оценочные средства для проведения промежуточного контроля

По окончанию практики студент должен представить руководителю практики отчет по практике, который включает:

1. разработанный и успешно протестированный программный продукт;
2. отчет в текстовом формате со следующей структурой (допускается в электронном виде):
  - титульный лист;
  - оглавление отчета;
  - введение;
  - разделы основной части отчета;
  - заключение;
  - список используемых источников;
  - приложения.

В отчете приводится информация общего характера (фамилия, имя, отчество студента; вид практики; период прохождения практики), указываются сведения о работе, выполнившейся бакалавром во время практики, отражаются результаты практики:

*Введение* – необходимо указать цель учебной-вычислительной практики, задачи, индивидуальное задание на практику

*Раздел* – описание выполнения заданий, полученных результатов.

*Заключение* – необходимо сформулировать выводы по результатам прохождения практики.

Содержание отчета по практике составляется в полном соответствии с индивидуальным заданием. Текст следует печатать через 1 межстрочный интервал с использованием шрифта Times New Roman, кегль 12, соблюдая следующие размеры полей: левое – 3 см, правое – 1,5 см, верхнее – 2 см, нижнее – 2 см. Абзацы в тексте следует начинать с отступа, равного 1,25 см.

### В) Вопросы к зачету по практике:

1. Современные технологии программирования.
2. Процесс разработки программного обеспечения.
3. Прикладное программное обеспечение.
4. Информационные системы.
5. Программные системы.
6. Алгоритмические языки программирования высокого уровня.

7. Интегрированная среда разработки Microsoft Visual Studio.
8. Структурное программирование.
9. Процедурное программирование.
10. Модульное программирование.
11. Объектно-ориентированное программирования.
12. Объекты и классы.
13. Абстракция. Иерархия.
14. Основные концепции объектно-ориентированного программирования.
15. Инкапсуляция. Наследование. Полиморфизм.
16. Консольные приложения. Разработка консольных приложений.
17. Язык программирования C#.
18. Структура программы на языке C#.
19. Элементы синтаксиса языка C#.
20. Типы данных языка C#.
21. Операторы языка C#.

Электронный адрес материалов на странице кафедры математики и информатики не предусмотрен.

Согласно реализуемой в Университете системе, учет и оценка знаний, умений и уровня сформированности компетенций у обучающегося осуществляется в два этапа:

1 этап: проведение текущего контроля успеваемости по практике, представляющий проверку усвоения учебного материала теоретического и практического характера, регулярно осуществляющуюся в процессе прохождения обучающимся практики.

2 этап: проведение промежуточной аттестации по итогам практики в последние три дня прохождения практики в форме зачета, зачета с оценкой, экзамена.

## Лист визирования

### Фонда оценочных материалов (оценочных средств)

Фонд оценочных материалов (оценочных средств) для проведения текущего контроля и промежуточной аттестации по учебной (производственной) практике Б2.О.01(У) Учебная практика технологическая проанализирован и признан актуальным для использования на 20\_\_\_\_-20\_\_\_\_ учебный год.

Протокол заседания кафедры \_\_\_\_\_  
наименование кафедры

от «\_\_\_\_» 20\_\_\_\_ г.

Зав. кафедрой

подпись

дата