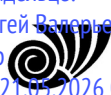


Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Таскаев Сергей Валерьевич
Должность: Ректор
Дата подписания: 21.05.2026 01:10:43
Уникальный программный ключ:
891934b8c2cf7b6350cbe51cdda3096e877f61f3



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ»)
Миасский филиал
Кафедра прикладной математики

Фонд оценочных средств по дисциплине «Современные технологии программирования»
по направлению подготовки 02.03.02 Фундаментальная информатика и информационные технологии, профиль
«Компьютерные науки» ФГБОУ ВО «ЧелГУ»

Версия документа - 1

стр. 1

Первый экземпляр _____

КОПИЯ № _____

Фонд оценочных средств для промежуточной аттестации

по дисциплине

Современные технологии программирования

Направление подготовки

02.03.02 Фундаментальная информатика и информационные технологии

Направленность (профиль)

Компьютерные науки

Присваиваемая квалификация
бакалавр

Форма обучения

очная

Миасс 2026 г.

**02.03.02 Фундаментальная информатика и информационные технологии,
Компьютерные науки, Современные технологии программирования, 2026, очная**

Фонд оценочных средств одобрен и рекомендован:

Проректор по учебной работе утверждено 27.02.26 А.А. Саламатов

Ученым советом Миасского филиала ФГБОУ ВО "ЧелГУ"

Протокол заседания № 8 от 24.02.2026

Председатель Ученого совета
Миасского филиала ФГБОУ ВО
"ЧелГУ"

согласовано

Т.В. Малькова

Заседанием кафедры прикладной математики

Протокол заседания № 6 от 30.01.2026

Заведующий кафедрой

согласовано

Е.В. Дутикова

Автор (составитель)

Д.Н. Дутиков

**Структура фонда оценочных средств соответствует приказу ректора ФГБОУ ВО
«ЧелГУ» от «13» апреля 2021 г. № 247-1**



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ»)
Миасский филиал
Кафедра прикладной математики

Фонд оценочных средств по дисциплине «Современные технологии программирования»
по направлению подготовки 02.03.02 Фундаментальная информатика и информационные технологии, профиль
«Компьютерные науки» ФГБОУ ВО «ЧелГУ»

Версия документа - 1

стр. 3 из 17

Первый экземпляр _____

КОПИЯ № _____

1. ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ

Направление подготовки 02.03.02 «Фундаментальная информатика и информационные технологии»

Направленность (профиль): Компьютерные науки.

Дисциплина: Современные технологии программирования

Семестр изучения: 7

Форма промежуточной аттестации: зачет.

2. ПЕРЕЧЕНЬ ФОРМИРУЕМЫХ КОМПЕТЕНЦИЙ

2.1. Компетенции, закреплённые за дисциплиной

Изучение дисциплины «Современные технологии программирования» направлено на формирование следующих компетенций:

Коды компетенции (по ФГОС)	Содержание компетенций согласно ФГОС	Индикаторы достижений	Перечень планируемых результатов обучения по дисциплине
1	2	3	4
ПК-2	Способен к разработке программного обеспечения, осуществлению интеграции программных модулей и компонент и проверке работоспособности программного обеспечения на основе международных и профессиональных стандартов в области	ПК-2.1 Демонстрирует знание основных принципов и технологий разработки программного обеспечения, методов и средств сборки модулей и компонент программного обеспечения; разработки процедур для развертывания программного обеспечения, методов и средств миграции и преобразования данных, методов создания пользовательских интерфейсов; средств программирования ПК-2.2 Демонстрирует умения разрабатывать	Знать: основные принципы объектно-ориентированного программирования: инкапсуляция, наследование, полиморфизм, а также назначение и основные принципы реализации общепринятых шаблонов проектирования Уметь: использовать принципы объектно-ориентированного программирования при разработке приложений, реализовывать общепринятые шаблоны проектирования и использовать их в приложениях Владеть:



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ»)
Миасский филиал
Кафедра прикладной математики

Фонд оценочных средств по дисциплине «Современные технологии программирования»
по направлению подготовки 02.03.02 Фундаментальная информатика и информационные технологии, профиль
«Компьютерные науки» ФГБОУ ВО «ЧелГУ»

Версия документа - 1

стр. 4 из 17

Первый экземпляр _____

КОПИЯ № _____

	информационных технологий	программный код на языках программирования высокого и низкого уровня, осуществлять отладку программ, оформлять техническую документацию; использовать выбранную среду программирования для разработки процедур интеграции программных модулей, проводить оценку работоспособности программного обеспечения ПК-2.3 Имеет практический опыт разработки исходного кода, тестирования программного обеспечения, сборки модулей и компонент программного обеспечения, разработки процедур для развертывания программного обеспечения, миграции и преобразования данных, создания программных интерфейсов	владеть навыками реализации приложений с помощью объектно-ориентированного подхода, а также навыками применения общепринятых шаблонов проектирования приложений
--	---------------------------	---	---

3. СОДЕРЖАНИЕ ОЦЕНОЧНЫХ СРЕДСТВ ПО ДИСЦИПЛИНЕ

3.1 Виды оценочных средств

№ п/п	Контролируемые темы/разделы	Код компетенции/ планируемые результаты обучения	Наименование оценочного средства для текущего контроля	Наименование оценочного средства на промежуточной аттестации
1.	Объектно-ориентированное программирование	ПК-2: Знает Понятие классов, наследо-	Лабораторная работа	Устный опрос (зачет)



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ»)
Миасский филиал
Кафедра прикладной математики

Фонд оценочных средств по дисциплине «Современные технологии программирования»
по направлению подготовки 02.03.02 Фундаментальная информатика и информационные технологии, профиль
«Компьютерные науки» ФГБОУ ВО «ЧелГУ»

Версия документа - 1

стр. 5 из 17

Первый экземпляр _____

КОПИЯ № _____

		<p>вания, полиморфизма, виртуальных функций, перегрузки операторов, шаблонов функций и классов</p> <p>Умеет Создавать классы, выполнять наследование классов для управления кодом, применять шаблоны для создания классов</p> <p>Владеет Навыками создания классов, применения наследования классов, применения шаблонов классов и функций</p>		
2.	Шаблоны проектирования	<p>ПК-2 Знает Назначение шаблонов Singleton, Factory Method, Composite, Command Умеет Реализовывать шаблоны Singleton, Factory Method, Composite, Command на языке C++ Владеет навыками применения шаблонов проектирования Singleton, Composite, Factory Method, Command</p>	Лабораторная работа	Устный опрос (зачет)

Типовые задания, критерии и показатели оценивания в рамках текущего контроля представлены в рабочей программе по дисциплине. Полные комплекты оценочных средств и контрольно-измерительных материалов хранятся на кафедре и являются учебно-методическими материалами ограниченного (конфиденциального) пользования.

3.2 Содержание оценочных средств для текущей аттестации

3.2.1 Лабораторные работы

Лабораторная работа №1

Цель работы: изучить понятие наследования и виртуальных методов,



научиться использовать виртуальные методы при использовании наследования.

Реализовать класс Database, имеющий следующий интерфейс:

```
class Database {
    vector<Person*> data;

    Database(string filename);

public:
    void print();
    void addRecord();
    void removeRecord(int index);
    void editRecord(int index);
};
```

Конструктор класса должен открыть файл с именем filename, содержащий данные в формате

3

```
{name: <Ivan>, surname: <Ivanov>, age: <15>, class_number: <9>, class_letter: <A>}
{name: <Petr>, surname: <Petrov>, age: <35>, discipline: <inf>}
{name: <Ivan>, surname: <Petrov>, age: <17>, class_number: <11>, class_letter: <B>}
```

Первая строка файла – количество записей, в остальные строки – это записи о ученике или преподавателе. Конструктор должен считать эти строки, определить тип записи (преподаватель или ученик), создать объект соответствующего класса, после чего добавить указатель на объект в массив data.

Предусмотрите генерацию исключений при отсутствии файла, а также при ошибках в расположении скобок внутри считываемого выражения.


Метод print выводит данные на печать в определенном формате:

Номер Фамилия Имя Возраст (Номер класса Буква класса / Дисциплина)

Например:

1. Ivanov Ivan 15 9A
2. Petrov Petr 35 inf
3. Petrov Ivan 17 11B

Номер записи соответствует ячейке массива data. Остальные данные выводятся с помощью методов потомков класса Person.

	МИНОБРНАУКИ РОССИИ Федеральное государственное бюджетное образовательное учреждение высшего образования «Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ») Миасский филиал Кафедра прикладной математики		
	Фонд оценочных средств по дисциплине «Современные технологии программирования» по направлению подготовки 02.03.02 Фундаментальная информатика и информационные технологии, профиль «Компьютерные науки» ФГБОУ ВО «ЧелГУ»		
Версия документа - 1	стр. 7 из 17	Первый экземпляр _____	КОПИЯ № _____

Метод `addRecord` считывает данные с клавиатуры и создаёт на основании этих данных объект типа `Student` или `Teacher`. Формат считывания данных и способ принятия решения о типе объекта определите самостоятельно.

Метод `removeRecord` удаляет запись из массива `data`. Помните, что при удалении указателя из массива нужно удалить и объект, связанный с указателем, с помощью оператора `delete`.

Метод `editRecord` изменяет данные объекта. При обновлении данных предусмотрите предложения для ввода, чтобы было понятно, какие именно данные обновляются.

Предусмотрите генерирование исключений при вводе неправильных данных. Например, при вводе отрицательного возраста должно генерироваться исключение с соответствующим сообщением об ошибке.

Класс `Person` – это базовый класс для классов `Student` и `Teacher`. Эти классы дополняют логику поведения класса `Person` за счет использования виртуальных методов.

```
class Person {
    string name;
    string surname;
    unsigned short age;

public:
    Person(string name, string surname, unsigned short age);
    virtual void updateData();
    virtual string print();
    virtual string printFile();
};
```

Метод `updateData` выполняет обновление данных, содержащихся в текущем объекте класса. Перегрузка этого метода в классах-наследниках выполняет обновление только полей класса, связанных с текущим классом.

Метод `print` формирует строку в определенном формате с данными `name`, `surname`, `age`. Данный метод в классах потомков дополняет строку данными этих классов.

Метод `printFile` выполняет формирование строки в формате файла, как было представлено выше:

```
{name: <Ivan>, surname: <Ivanov>, age: <15>, class_number: <9>, class_letter: <A>}
```



Классы потомки дополняют эту строку своими данными.

Лабораторная работа №2

Цель работы: изучить принципы работы абстрактных типов данных стек, научиться использовать шаблоны классов для реализации средств обобщенного программирования.

Задание 1. Простой стек

Реализуйте структуру данных "стек", реализовав все указанные здесь методы. Напишите программу (функцию main), содержащую описание стека и моделирующую работу стека. Функция main считывает последовательность команд и в зависимости от команды выполняет ту или иную операцию. После выполнения одной команды программа должна вывести одну строчку. Возможные команды для программы:

push n

Добавить в стек число n (значение n задается после команды). Программа должна вывести ok.

pop

Удалить из стека последний элемент. Программа должна вывести его значение.

back

Программа должна вывести значение последнего элемента, не удаляя его из стека.

size

Программа должна вывести количество элементов в стеке.

clear

Программа должна очистить стек и вывести ok.

exit

Программа должна вывести bye и завершить работу.

Гарантируется, что набор входных команд удовлетворяет следующим требованиям: максимальное количество элементов в стеке в любой момент не превосходит 100, все команды pop_back и back корректны, то есть при их исполнении в стеке содержится хотя бы один элемент.

Пример протокола работы программы

Ввод	Вывод
-------------	--------------

push 2	ok
push 3	ok



```
push 5      ok
back        5
size        3
pop         5
size        2
push 7      ok
pop         7
clear       ok
size        0
exit        bye
```

Задание 2. Стек с обработкой ошибок

Аналогично предыдущему заданию, только снимается ограничение на корректность вызовов методов `back` и `pop`. Данные операции должны перед исполнением проверять, содержится ли в стеке хотя бы один элемент. Если во входных данных встречается операция `back` или `pop`, при этом стек пуст, то программа должна вместо числового значения вывести строку `error`.

При этом должна быть реализована двойная защита: вызов методов `forward` и `pop` для пустого стека не должен приводить к обращению к несуществующим элементам массива `m_elems`, а функция `main` должна выводить сообщение `error`, при считывании некорректной операции.

Пример протокола работы программы

Ввод	Вывод
push 2	ok
back	2
pop	2
size	0
pop	error
push 1	ok
size	1
exit	bye

Задание 3. Шаблонный класс стека

Создайте шаблонный класс стека. Шаблон класса должен принимать тип данных, с которым работает данный экземпляр стека. Модифицируйте код из предыдущей задачи, чтобы описанный вами стек мог бы работать с произвольными типами данных.



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ»)
Миасский филиал
Кафедра прикладной математики

Фонд оценочных средств по дисциплине «Современные технологии программирования»
по направлению подготовки 02.03.02 Фундаментальная информатика и информационные технологии, профиль
«Компьютерные науки» ФГБОУ ВО «ЧелГУ»

Версия документа - 1

стр. 10 из 17

Первый экземпляр _____

КОПИЯ № _____

Лабораторная работа №3

Цель работы: изучить шаблон разработки Singleton, реализовать взаимодействие между компонентами с помощью шаблона Singleton.

В качестве проекта выберем реализацию игры тетрис.

Начнем реализацию проекта с создания класса, который описывает каждое отдельное тетрамино. Для отображения и управления положением фигурки будем класс со следующим интерфейсом:

```
typedef std::vector<sf::Vector2i> tetramino_schemes;

class Tetramino {
    unsigned short rotationCenter;
    bool rotatable;

    std::vector<sf::Sprite> sprites;

    bool isOutOfField();

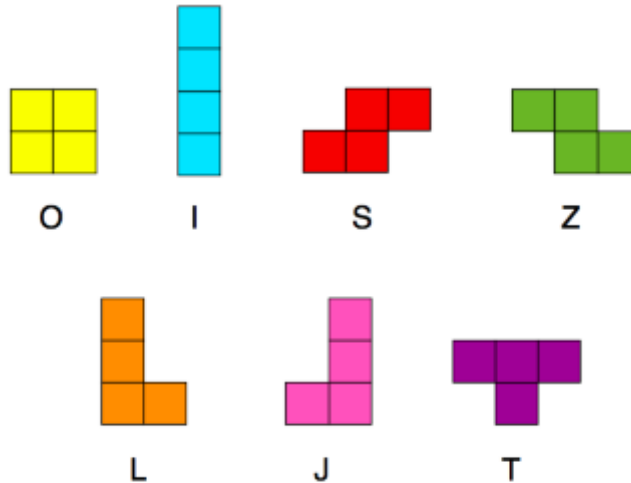
public:
    Tetramino(unsigned short type, short x, short y,
tetramino_schemes scheme, unsigned short rotationCenter, bool
isRotatable);

    void moveLeft();
    void moveRight();
    void rotateRight();
    void rotateLeft();
    void tick();
    bool isOnLand(unsigned int fieldHeight);

    size_t getSpritesCount();

    sf::Sprite getSprite(int number);
};
```

Всего же в Тетрисе используется 7 фигурок-тетрамино



Каждая такая фигура будет описываться с помощью вектора `tetramino_schemes`. С помощью данного класса можно управлять положением одной фигуры, а точнее набора спрайтов, которые соответствуют этой фигуре. Опишем более подробно методы класса:

`Tetramino` – конструктор класса, он получает ряд аргументов, которые определяют тип спрайтов фигуры, точнее цвет этих спрайтов, положение на экране основного спрайта, от которого будут рассчитываться положения остальных спрайтов, тип тетрамино (как представлено на картинке выше), центр вращения фигуры, чтобы определить спрайт, относительно которого нужно выполнять поворот, параметр, который определяет можно ли вообще вращать данную фигуру.

`moveLeft` – выполняет сдвиг фигуры влево, если это возможно.

`moveRight` – выполняет сдвиг фигуры вправо, если это возможно.

`rotateRight` – выполняет поворот фигуры на +90 градусов.

`rotateLeft` – выполняет поворот фигуры на -90 градусов.

`tick` – выполняет перемещение фигуры вниз на одну позицию.

`isOnLand` – определяет коснулась ли фигуры нижней границы игрового поля.

`getSpritesCount` – возвращает количество спрайтов в фигуре.

`getSprite` – возвращает спрайт под позицией `number`.

Для получения текстуры для спрайтов, а также шрифтов, аудио фрагментов и так далее будем использовать класс `AssetManager`. Данный класс будет реализован при помощи шаблона проектирования `Singleton`, так как данный класс всегда должен быть представлен всегда в одном экземпляре в приложении. С другой стороны передавать этот класс по ссылке во все



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ»)
Миасский филиал
Кафедра прикладной математики

Фонд оценочных средств по дисциплине «Современные технологии программирования»
по направлению подготовки 02.03.02 Фундаментальная информатика и информационные технологии, профиль
«Компьютерные науки» ФГБОУ ВО «ЧелГУ»

Версия документа - 1

стр. 12 из 17

Первый экземпляр _____

КОПИЯ № _____

требуемые точки приложения не очень удобно. Так как назначение этого класса сходно с хранением, например, конфигурации приложения, то возможно его реализовать с помощью данного шаблона разработки.

Интерфейс класса, который реализует такую логику поведения, можно представить, например, следующим образом:

```
class AssetManager {
    std::map<std::string, sf::Texture> textures;
    std::map<std::string, sf::Font> fonts;
    AssetManager();
    AssetManager& operator = (const AssetManager& manager);
    AssetManager(const AssetManager& manager);

public:
    static AssetManager* getManager();

    void loadTexture(const std::string& name, const std::string&
filename);
    sf::Texture& getTexture(const std::string& name);

    void loadFont(const std::string& name, const std::string&
filename);
    sf::Font& getFont(const std::string& name);
};
```

В данном классе за счет строк

```
AssetManager();
AssetManager& operator = (const AssetManager& manager);
AssetManager(const AssetManager& manager);
```

Запрещается использовать конструктор по умолчанию, конструктор копирования и оператор присваивания за пределами класса. Чтобы получать возможность работать с объектом класса вводится статический метод

```
static AssetManager* getManager();
```

который позволяет получить указать на объект этого класса. Такая конструкция позволяет гарантировать то, что в отдельный момент времени в программе будет только один объект класса AssetManager.

3.3. Критерии оценивания по видам оценочных средств

Критерии оценивания лабораторной работы

Оценка	Неудовлетворительно	Удовлетворительно	Хорошо	Отлично
Критерии	Студент не выполнил задание: не представил	Студент правильно	Студент правильно	Студент правильно



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ»)
Миасский филиал
Кафедра прикладной математики

Фонд оценочных средств по дисциплине «Современные технологии программирования»
по направлению подготовки 02.03.02 Фундаментальная информатика и информационные технологии, профиль
«Компьютерные науки» ФГБОУ ВО «ЧелГУ»

Версия документа - 1

стр. 13 из 17

Первый экземпляр _____

КОПИЯ № _____

	алгоритм решения задачи, не составил исходный код программной реализации, программное решение некорректно работает для некоторых значений входных параметров студент не может объяснить предложенное решение	выполнил задание к лабораторной работе. Составил отчет в установленной форме. Привел исходный код решения, программное решение работоспособно. Студент может объяснить предложенное решение	выполнил задание к лабораторной работе. Составил отчет в установленной форме. Привел исходный код решения, программное решение работоспособно. Студент может объяснить предложенное решение. Студент способен ответить на теоретические вопросы, испытывая небольшие затруднения.	выполнил задание к лабораторной работе. Составил отчет в установленной форме. Привел исходный код решения, программное решение работоспособно. Студент может объяснить предложенное решение. Студент способен ответить на теоретические вопросы.
--	--	---	---	--

4. ПОРЯДОК ПРОВЕДЕНИЯ И КРИТЕРИИ ОЦЕНИВАНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

4.1 Порядок проведения и содержание оценочных средств для промежуточной аттестации

Для получения зачета студенту необходимо сдать все лабораторные работы, а также ответить на теоретические вопросы билета, выбранного случайным образом. Время выполнения – 60 минут.

Перечень вопросов к зачету:

1. Что такое объектно-ориентированное программирование?
2. Что означает термин "Инкапсуляция"?
3. Что означает термин "Наследование"?
4. Что означает термин "Полиморфизм"?
5. Дайте определения следующих терминов: класс; объект; поведение. Как объекты обмениваются информацией?
6. Каким образом использование инкапсуляции помогает достичь целей объектно-ориентированного программирования?
7. Дайте определение понятию "абстракция" и приведите пример применения абстракции. Определите понятие типа. Что такое абстрактный тип данных?



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ»)
Миасский филиал
Кафедра прикладной математики

Фонд оценочных средств по дисциплине «Современные технологии программирования»
по направлению подготовки 02.03.02 Фундаментальная информатика и информационные технологии, профиль
«Компьютерные науки» ФГБОУ ВО «ЧелГУ»

Версия документа - 1

стр. 14 из 17

Первый экземпляр _____

КОПИЯ № _____

8. Дайте определение понятию "реализация". Дайте определение понятию "интерфейс". Объясните разницу между интерфейсом и реализацией.
9. Почему для достижения эффективной инкапсуляции важно четко распределить ответственность?
10. Как можно получить эффективное сокрытие реализации в программе? Какие опасности заключает в себя абстракция?
11. Назовите и объясните три формы наследования.
12. Каким образом использование наследования помогает достичь целей объектно-ориентированного программирования?
12. Назовите причины использования шаблона разработки Singleton.
13. Опишите принцип реализации шаблона Singleton.
14. Назовите причины использования шаблона разработки Factory Method.
15. Опишите принцип реализации шаблона Factory Method.
16. Назовите причины использования шаблона разработки Composite.
17. Опишите принцип реализации шаблона Composite.
18. Назовите причины использования шаблона разработки Command.
19. Опишите принцип реализации шаблона Command.

4.2 Критерии оценивания компетенций в ходе промежуточной аттестации

Код компетенции	Планируемые результаты обучения по дисциплине	Критерии оценивания	
		Не зачтено	Зачтено
ПК-2	Знает понятие классов, наследования, полиморфизма, виртуальных функций, перегрузки операторов, шаблонов функций и классов, назначение шаблонов Singleton, Factory Method, Composite, Command	Не знает понятие классов, наследования, полиморфизма, виртуальных функций, перегрузки операторов, шаблонов функций и классов, назначение шаблонов Singleton, Factory Method, Composite, Command	Знает понятие классов, наследования, полиморфизма, виртуальных функций, перегрузки операторов, шаблонов функций и классов, назначение шаблонов Singleton, Factory Method, Composite, Command, но допускает незначительные погрешности
	Умеет создавать классы, выполнять наследование классов для управ-	Не умеет создавать классы, выполнять наследование классов для управления кодом, приме-	Умеет создавать классы, выполнять наследование классов для управления кодом, применять шаблоны



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ»)
Миасский филиал
Кафедра прикладной математики

Фонд оценочных средств по дисциплине «Современные технологии программирования»
по направлению подготовки 02.03.02 Фундаментальная информатика и информационные технологии, профиль
«Компьютерные науки» ФГБОУ ВО «ЧелГУ»

Версия документа - 1

стр. 15 из 17

Первый экземпляр _____

КОПИЯ № _____

ления кодом, применять шаблоны для создания классов, реализовывать шаблоны Singleton, Factory Method, Composite, Command на языке C++	нять шаблоны для создания классов, реализовывать шаблон Singleton, Factory Method, Composite, Command на языке C++.	для создания классов, реализовывать шаблоны Singleton, Factory Method, Composite, Command на языке C++
Владеет навыками создания классов, применения наследования классов, применения шаблонов классов и функций, владеет навыками применения шаблонов проектирования Singleton, Composite, Factory Method, Command	Не владеет навыками создания классов, применения наследования классов, применения шаблонов классов и функций, владеет навыками применения шаблонов проектирования Singleton, Composite, Factory Method, Command	Владеет навыками создания классов, применения наследования классов, применения шаблонов классов и функций, владеет навыками применения шаблонов проектирования Singleton, Composite, Factory Method, Command

4.3 Критерии оценивания зачета.

Критериями ответа на зачёте выступают следующие качества знаний:

полнота – количество знаний об изучаемом объекте, входящих в программу;

глубина – совокупность осознанных знаний об объекте;

конкретность – умение раскрыть конкретные проявления обобщённых знаний (доказать на примерах основные положения);

системность – представление знаний об объекте в системе, с выделением структурных её элементов, расположенных в логической последовательности;

развёрнутость – способность развернуть знания в ряд последовательных шагов;

осознанность – понимание связей между знаниями, умение выделить существенные и несущественные связи, познание способов и принципов получения знаний.

Ответ студента по вопросам дисциплины оценивается положительно с выставлением оценки «зачтено» в следующих случаях:



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ»)
Миасский филиал
Кафедра прикладной математики

Фонд оценочных средств по дисциплине «Современные технологии программирования»
по направлению подготовки 02.03.02 Фундаментальная информатика и информационные технологии, профиль
«Компьютерные науки» ФГБОУ ВО «ЧелГУ»

Версия документа - 1

стр. 16 из 17

Первый экземпляр _____

КОПИЯ № _____

- студент глубоко и полно владеет содержанием учебного материала; умеет связывать теорию с практикой, решает соответствующие задачи, теоретические выводы подтверждает примерами, фактами, данными научных исследований; осуществляет межпредметные связи, предложения. Делает выводы логично, четко. Ясно и кратко излагает ответы на поставленные вопросы; умеет обосновывать свои суждения и профессионально-личностную позицию по излагаемому вопросу. Дан полный, развёрнутый ответ на поставленный вопрос; показана совокупность осознанных знаний об объекте изучения, доказательно раскрыты основные положения (свободно оперирует понятиями и терминами); в ответе прослеживается чёткая структура, выстроенная в логической последовательности; ответ изложен литературным грамотным языком и носит самостоятельный характер.

– ответ студента соответствует указанным выше критериям, но содержание ответа имеет отдельные неточности (несущественные ошибки) в изложении теоретического и практического материала, отличается меньшей обстоятельностью, глубиной, обоснованностью и полнотой; были допущены неточности в определении понятий и терминов, допущенные ошибки исправляются студентом после дополнительных вопросов преподавателя.

– студент обнаруживает знание и понимание основных положений учебного материала, но излагает его неполно, непоследовательно, допускает неточности и существенные ошибки в определении понятий, формулировке положений, не привлекает для аргументации ответа основные положения концептуальных и нормативных документов, не умеет обосновать свои суждения; наблюдается нарушение логики изложения; в ответе не присутствуют доказательные выводы; сформированность умений показана слабо.

Оценка «не зачтено» за устный ответ студента по вопросам дисциплины выставляется в случаях, когда:

– студент имеет разрозненные, бессистемные знания: не умеет выделять главное и второстепенное; допускает ошибки в определении понятий, формулировке теоретических положений, искажает их смысл; не ориентируется в нормативно-концептуальных, программно-методических, исследовательских материалах, беспорядочно и неуверенно излагает материал; не умеет соединять теоретические положения с практикой; не умеет применять знания для обоснования и объяснения фактов, не устанавливает межпредметные связи.

При необходимости инвалидам и лицам с ограниченными возможностями здоровья предоставляется дополнительное время для подготовки ответа на зачете.



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ»)
Миасский филиал
Кафедра прикладной математики

Фонд оценочных средств по дисциплине «Современные технологии программирования»
по направлению подготовки 02.03.02 Фундаментальная информатика и информационные технологии, профиль
«Компьютерные науки» ФГБОУ ВО «ЧелГУ»

Версия документа - 1

стр. 17 из 17

Первый экземпляр _____

КОПИЯ № _____

4.4. Результаты промежуточной аттестации и уровни сформированности компетенций

Уровень освоения компетенций	Оценка
Продвинутый	зачтено
Базовый	зачтено
Пороговый	зачтено
компетенции не сформированы	не зачтено

Уровни формирования компетенций:

1. Пороговый уровень:

- предполагает формирование компетенций на начальном уровне: знание основ языка программирования C++;
- студент способен давать ответы на теоретические вопросы дисциплины на удовлетворительном уровне.

2. Базовый уровень:

- предполагает формирование компетенций на более высоком уровне: формируется комплексное знание особенностей и применения методов языка программирования C++;
- студент способен давать развернутые ответы на теоретические вопросы дисциплины; способен решать практические задания.

3. Продвинутый уровень:

- предполагает формирование компетенций на высоком уровне, использует полученные знания и умения при изучении смежных дисциплин, обнаруживает готовность к самостоятельной профессиональной деятельности;
- студент способен аргументировать собственную точку зрения, формулировать собственные выводы на основе применения усвоенных компетенций.