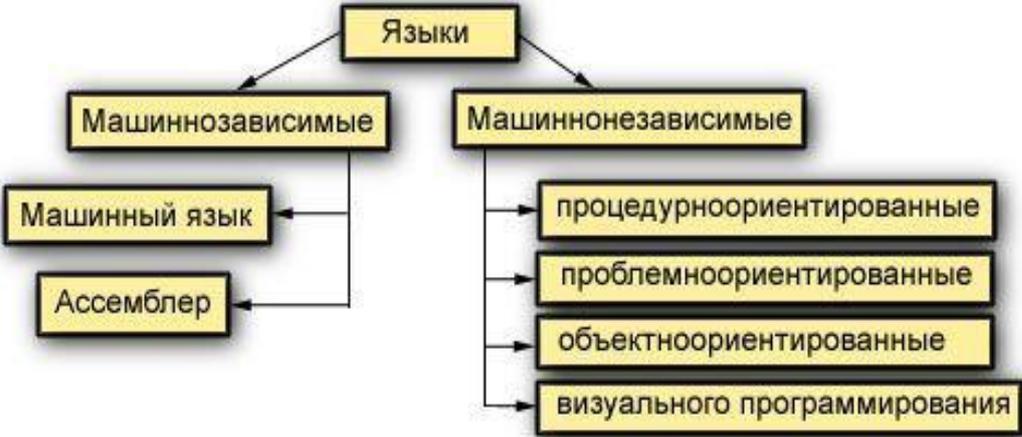


Информация для размещения на официальном сайте ГБПОУ «Светлоградский
региональный сельскохозяйственный колледж»

Для электронного обучения

Группа	321
Дата	27.02.2025
Время	11.50 – 13.10
Наименование УД/МДК/УП/ПП	МДК 01.04 Системное программирование
Ф.И.О. преподавателя	Коваленко Аркадий Владимирович
Электронная почта	Aricus2007@inbox.ru
Основная литература	<p>Основные источники:</p> <ol style="list-style-type: none"> 1. Разработка модулей программного обеспечения для компьютерных систем. Г. Н. Федорова из. Академия 2024 год https://book.ru/books/957489 2. Тищенко В.И. Учебное пособие по курсу «Системное программирование»; 3. Тищенко В.И. Лабораторный практикум по курсу «Системное программирование»; 4. Тищенко В.И. Лабораторный практикум «Разработка оверлейных и резидентных программ»; 5. Юров В. Ассемблер – учебник; 6. Юров В. Ассемблер – практикум.
Тема	Введение. Классификация языков программирования. Машинный язык и язык ассемблера. Структурная схема компьютера.
Задание(лекция)	<p>Классификация языков программирования</p>  <pre> graph TD A[Языки] --> B[Машиннозависимые] A --> C[Машиннонезависимые] B --> D[Машинный язык] B --> E[Ассемблер] C --> F[процедурноориентированные] C --> G[проблемноориентированные] C --> H[объектноориентированные] C --> I[визуального программирования] </pre> <p>Машинный язык программирования Машинный язык программирования – это язык, непосредственно воспринимаемый компьютером. Каждая его команда интерпретируется аппаратурой ЭВМ. Машинная команда имеет структуру вида: код операции, операнд1, операнд2 Операнды - это данные или адреса, над которыми будет выполняться действие, определенное кодом операции. Структура данных в оперативной памяти называется форматом данных.</p> <p>Ассемблер как язык программирования Языки ассемблерного типа используют мнемоническое обозначение адресов и кодов операций. Ассемблер включает в себя:</p>

- машинные команды,
- символические адреса,
- макросы,

комментарии.

Этапы развития языков программирования

Выделим 5 основных поколений:

1. (конец 50-х г.) – Fortran, Algol;
2. (середина 60-х г.) - Cobol, Lisp;
3. (70-е годы) - PL/1, Pascal;
4. (80-е годы) - Object Pascal, C++, Ada;
5. (90-е годы) - Visual Basic, Delphi, Builder.

Функциональные возможности языков и технологии программирования

- с развитием аппаратных средств появились: функции ввода-вывода, поддержка файловой системы, взаимодействие с операционной системой;

- с усложнением задач:

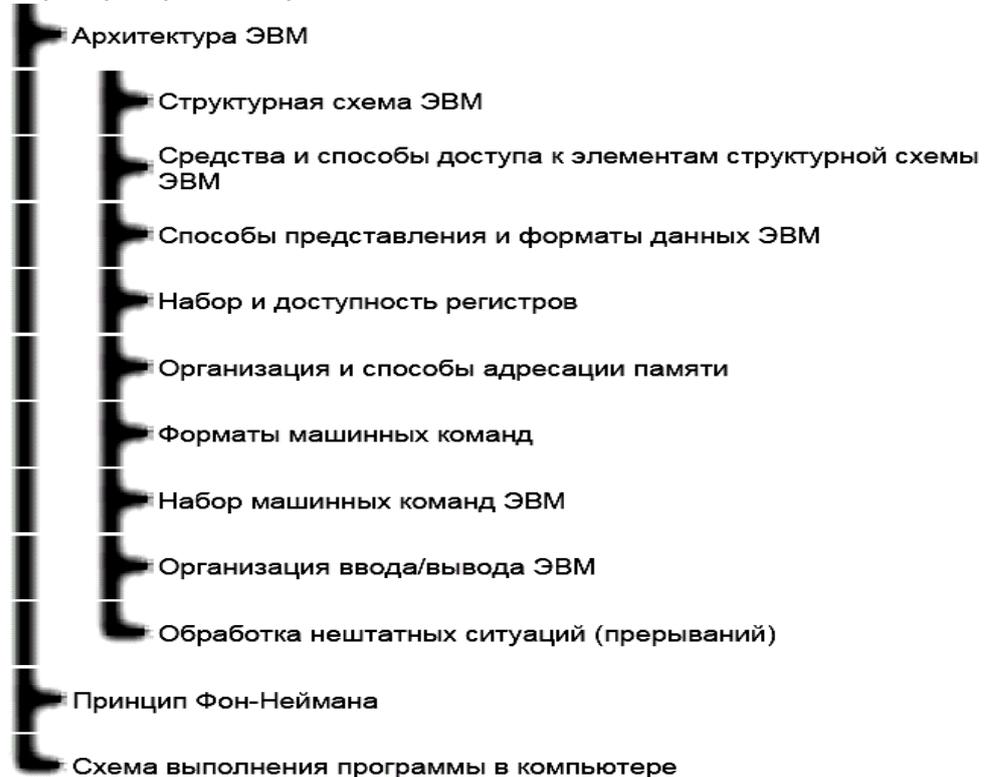
поддержка подпрограмм, механизм передачи параметров (основа для методологии **структурного программирования**);

- возможность создания больших программ на основе подпрограмм изменили архитектуру языков и подход к компоновке программ (**механизм раздельной трансляции программ и понятие модульности**);
- абстракция данных, типизация и модульность – основа **технологии объектно-ориентированного программирования**;
- появление среды Windows породило **технологии визуального программирования**.

Архитектура ЭВМ

Архитектура ЭВМ – это абстрактное представление ЭВМ, которое отражает ее структурную, схемотехническую и логическую организации.

Общие принципы построения ЭВМ



Структурная схема компьютера

	<p style="text-align: center;">Центральный процессор</p> <p>The diagram illustrates a computer system architecture. On the left, a large yellow box labeled 'Центральный процессор' (Central Processor) contains five sub-components: 'устройство управления' (Control Unit), 'блок микрокоманд' (Microcommand Block), 'регистровая память' (Register Memory), 'арифметико-логическое устройство' (Arithmetic Logic Unit), and 'кеш-память 1-го уровня' (Level 1 Cache Memory). Below this box is a separate yellow box for 'оперативная память' (Main Memory). On the right, several peripheral devices are shown in yellow boxes: 'видео' (Video), 'аудио' (Audio), 'НЖМД' (Hard Disk Drive), 'НГМД' (Optical Drive), 'принтер' (Printer), and 'модем' (Modem). Three vertical black lines represent system buses: the leftmost is the 'шина адреса' (Address Bus), the middle is the 'шина данных' (Data Bus), and the rightmost is the 'шина управления' (Control Bus). Bidirectional arrows connect the processor and memory to these buses, and unidirectional arrows connect the peripheral devices to the control bus. At the bottom, the text 'шины: адреса данных управления' (buses: address data control) is written.</p>
Закрепление знаний(контроль)	Сделать конспект лекции и предоставить 7.03.2025

Дата 27.02.2025

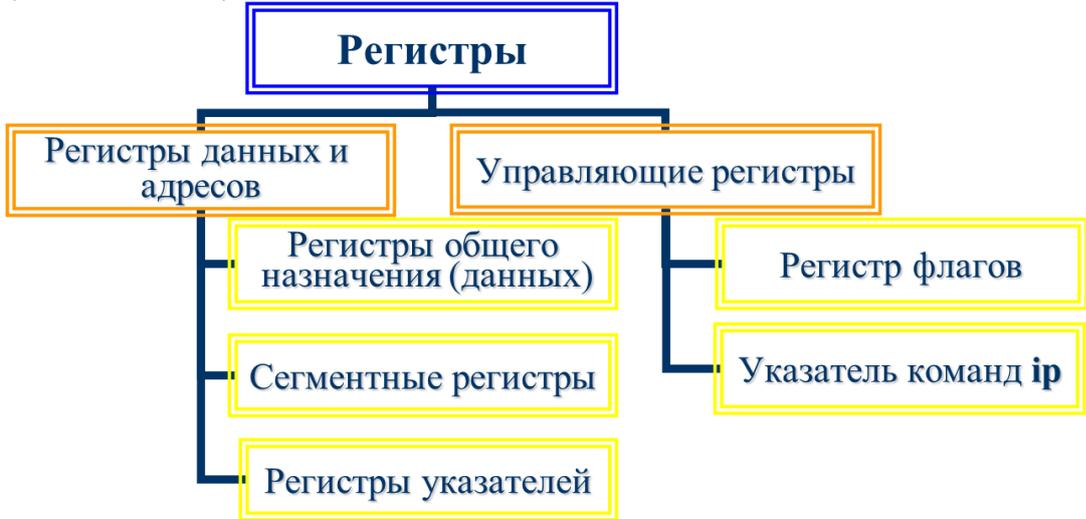
Подпись

Коваленко А. В.

Ф.И.О. преподавателя

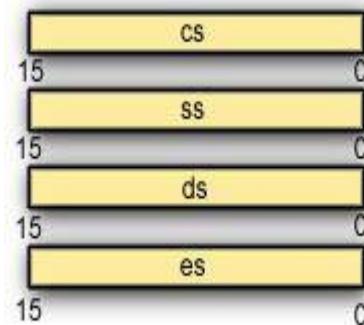
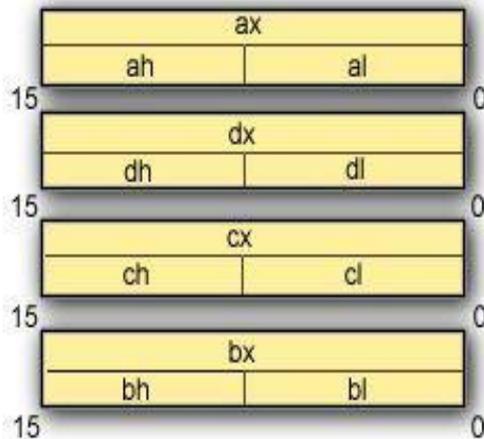
Информация для размещения на официальном сайте ГБПОУ «Светлоградский
региональный сельскохозяйственный колледж»

Для электронного обучения

Группа	321
Дата	28.02.2025
Время	11.50 – 13.10
Наименование УД/МДК/УП/ПП	МДК 01.04 Системное программирование
Ф.И.О. преподавателя	Коваленко Аркадий Владимирович
Электронная почта	Aricus2007@inbox.ru
Основная литература	<p>Основные источники:</p> <ol style="list-style-type: none"> 1. Разработка модулей программного обеспечения для компьютерных систем. Г. Н. Федорова из. Академия 2024 год https://book.ru/books/957489 2. Тищенко В.И. Учебное пособие по курсу «Системное программирование»; 3. Тищенко В.И. Лабораторный практикум по курсу «Системное программирование»; 4. Тищенко В.И. Лабораторный практикум «Разработка оверлейных и резидентных программ»; 5. Юров В. Ассемблер – учебник; 6. Юров В. Ассемблер – практикум.
Тема	Классификация регистров. Назначение регистров. Адресация памяти. Физическая адресация памяти.
Задание(лекция)	<p>Регистры Электронное устройство, предназначенное для временного хранения информации, называется регистром. Расположены на кристалле МП. Характеризуются размером: 8-разрядные, 16-разрядные, 32-разрядные, 64-разрядные.</p> <p>Регистры (intel 8086/8088) IBM/PC</p>  <pre> graph TD A[Регистры] --> B[Регистры данных и адресов] A --> C[Управляющие регистры] B --> D[Регистры общего назначения (данных)] B --> E[Сегментные регистры] B --> F[Регистры указателей] C --> G[Регистр флагов] C --> H[Указатель команд ip] </pre>

Регистры данных Сегментные регистры

Регистры общего назначения:



Регистры индексов и указателей

- sp – указатель стека;
- bp – указатель базы;
- si – индекс источника;
- di – индекс результата.

Адрес памяти задается двумя значениями – сегмент и смещение, например : ds:dx, es:bx, ss:sp, ss:bp.

Текущая исполняемая команда определяется **cs:ip**.

Управляющие регистры

- ip – указатель команд,
- регистр флагов.

Указатель команд - содержит смещение следующей команды в кодовом сегменте.

Регистры 32- разрядного МП Pentium

- 16 – системных;
- 16 – пользовательских:
 - 8 РОИ, индексных и указателей по 32 разряда **eax/ax/ah/al**, **ebx/bx/bh/bl**, **edx/dx/dh/dl**, **ecx/cx/ch/cl**,
 - 6 сегментных по 16 разрядов: cs, ds, ss, es, fs, gs;
 - 2 регистра состояния и управления по 32 разряда: **eip/ip** и **eflags/ flags**.

Адресация памяти

Адресация памяти

Физическая

В ассемблере

Способы адресации

Регистровая

Непосредственная

Прямая

Косвенная регистровая

Адресация по базе

Прямая с индексированием

Адресация по базе с индексированием

Некоторые константы

16 32 20 24
2¹⁶ = 64 кб, 2³² = 4Гб, 2²⁰ = 1Мб, 2²⁴ = 16 Мб.

Для МП intel 8088 размер машинного слова –
16 бит или 2 байта, шина имела 20 линий,
поэтому адрес 20- разрядный.

Физическая адресация памяти

Адрес, выдаваемый на шину адреса, называется **физическим**.

Физический адрес = (сегментный адрес)*16 + смещение
или

Физический адрес = (сегментный адрес)*10h + смещение.

Пример вычисления физ. Адреса

Пусть содержимое сегментного регистра равно 2011h, смещение равно 15h,
тогда ФА=20110h+15h= 20125h

Расположение машинного слова в памяти

Младший байт записывается в ячейку с меньшим адресом, старший – в ячейку с адресом на 1 больше.

Пример:

пусть число 1234h размещено с адреса 1927:0000, т.е. занимает ф.а. 19270h и 19271h.

Тогда цифры 34h – по адресу 19270h,
а 12h – по адресу 19271h.

Назначение регистров – РОН

АХ – аккумулятор.

ВХ – как вычислительный регистр, но может быть адресным.

СХ – счетчик в некоторых командах.

ДХ – расширитель аккумулятора.

Назначение регистров адресации

si, di, bp, bx – основное назначение – хранить 16 – разрядное значение при формировании адреса.

Назначение регистров - управляющие регистры

- **ip** – указатель команд,
- **регистр флагов.**

Указатель команд - содержит смещение следующей команды.

	<p>Методы изменения порядка выполнения команд:</p> <ol style="list-style-type: none"> 1. последовательный порядок команд, 2. переход внутри сегмента (near – переход), 3. переход в другой сегмент (far – переход). <p>Указатель стека sp Определяет смещение текущей вершины стека. Адрес стека определяется как ss:sp или ss:bp. Пример загрузки сегментных регистров cs в ds:</p> <p>а) mov ax, cs mov ds, ax</p> <p>в) push cs pop ds</p>
<p>Закрепление знаний(контроль)</p>	<p>Сделать конспект лекции и предоставить 7.03.2025</p>

Дата 28.02.2025

Подпись

Коваленко А. В.,

Ф.И.О. преподавателя

Информация для размещения на официальном сайте ГБПОУ «Светлоградский
региональный сельскохозяйственный колледж»

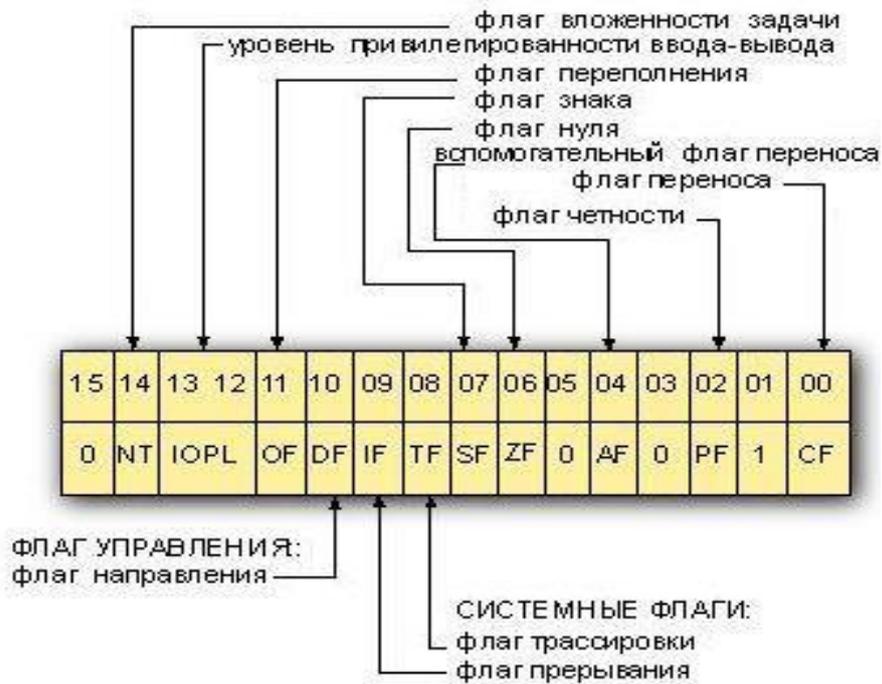
Для электронного обучения

Группа	321																						
Дата	28.02.2025																						
Время	13.20 – 14.40																						
Наименование УД/МДК/УП/ПП	МДК 01.04 Системное программирование																						
Ф.И.О. преподавателя	Коваленко Аркадий Владимирович																						
Электронная почта	Aricus2007@inbox.ru																						
Основная литература	<p>Основные источники:</p> <p>7. Разработка модулей программного обеспечения для компьютерных систем. Г. Н. Федорова из. Академия 2024 год https://book.ru/books/957489</p> <p>8. Тищенко В.И. Учебное пособие по курсу «Системное программирование»;</p> <p>9. Тищенко В.И. Лабораторный практикум по курсу «Системное программирование»;</p> <p>10. Тищенко В.И. Лабораторный практикум «Разработка оверлейных и резидентных программ»;</p> <p>11. Юров В. Ассемблер – учебник;</p> <p>12. Юров В. Ассемблер – практикум.</p>																						
Тема	Регистр флагов. Механизм формирования физического адреса. Форматы данных. Директивы определения данных.																						
Задание(лекция)	<p><i>Содержимое регистра flags</i></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">имя флага</th> <th style="text-align: center;">1</th> <th style="text-align: center;">0</th> </tr> </thead> <tbody> <tr> <td>переполнение (да/нет) overflow</td> <td style="text-align: center;">OV</td> <td style="text-align: center;"><u>NV</u></td> </tr> <tr> <td>направление (↑/↓) directory</td> <td style="text-align: center;">DN</td> <td style="text-align: center;"><u>UP</u></td> </tr> <tr> <td>прерывание (разрешено/запрещено)</td> <td style="text-align: center;"><u>EI</u></td> <td style="text-align: center;">DI</td> </tr> <tr> <td>знак (-/+) sign</td> <td style="text-align: center;">NG</td> <td style="text-align: center;"><u>PL</u></td> </tr> <tr> <td>нуль (да/нет) zero</td> <td style="text-align: center;">ZR</td> <td style="text-align: center;"><u>NZ</u></td> </tr> <tr> <td>дополнительный перенос(да/нет)</td> <td style="text-align: center;">AC</td> <td style="text-align: center;"><u>NA</u></td> </tr> </tbody> </table>		имя флага	1	0	переполнение (да/нет) overflow	OV	<u>NV</u>	направление (↑/↓) directory	DN	<u>UP</u>	прерывание (разрешено/запрещено)	<u>EI</u>	DI	знак (-/+) sign	NG	<u>PL</u>	нуль (да/нет) zero	ZR	<u>NZ</u>	дополнительный перенос(да/нет)	AC	<u>NA</u>
имя флага	1	0																					
переполнение (да/нет) overflow	OV	<u>NV</u>																					
направление (↑/↓) directory	DN	<u>UP</u>																					
прерывание (разрешено/запрещено)	<u>EI</u>	DI																					
знак (-/+) sign	NG	<u>PL</u>																					
нуль (да/нет) zero	ZR	<u>NZ</u>																					
дополнительный перенос(да/нет)	AC	<u>NA</u>																					

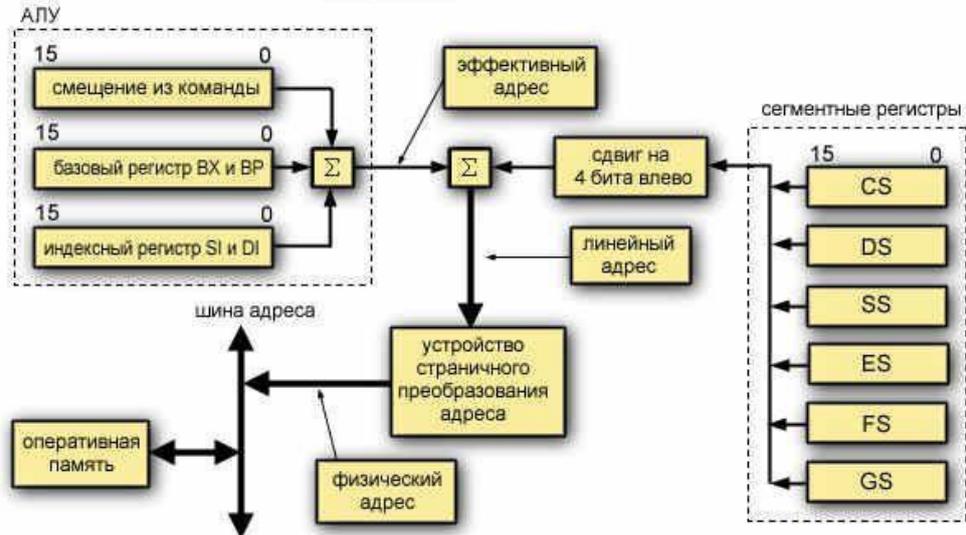
четность(чет/нечет) parity	PE	PO
перенос (есть/нет) carry	CY	NC

Обозначения регистров под отладчиком

ФЛАГИ СОСТОЯНИЯ:



Механизм формирования физического адреса в реальном режиме



Типы данных

Классификация данных по разрядности

(поддерживается МП на аппаратном уровне)

Типы арифметических данных: логическая структура

Форматы арифметических данных



Форматы данных сопроцессора

8 регистров данных длиной 80 бит.

Оперирует 7 типами данных:

- 3 типа целых (слово 16 бит, короткое целое 32 бита, длинное целое 64 бита);
- 3 типа вещественных (короткое 32 бита, длинное 64 бита, временное 80 бит):
знак, характеристика, мантисса
1 бит 8, 11 или 15 бит 23, 52 или 64 бита;
- Упакованные двоично-десятичные числа.

Операторы ассемблера

Общий вид оператора ассемблера:

метка КОП операнд_1, операнд_2

Оператором может быть:

- машинная команда;
- директива транслятора;
- макрокоманда;

комментарий.

Директивы транслятора

для определения данных в ассемблере

формат директивы:

имя	d	выражение
a	dw	10 ; в десятичной системе счисления
b	dw	10h ; использование шестнадцатеричной системы счисления

Выражение

в директиве определения данных

может быть:

- константой:
ABC1 dw 1234h
- списком:
ABC2 db 1,2,3
- строкой:
ABC3 db 'stroka'

	● с операцией дублирования: ABC4 db 4 dup (0)
Закрепление знаний(контроль)	Сделать конспект лекции и предоставить 7.03.2025

Дата 28.02.2025

Подпись

Коваленко А. В.,

Ф.И.О. преподавателя