

# Семестровый проект

Практические занятия

БИВТ-24-17

**Надежда Анисимова**

**ms teams m2102039@edu.misis.ru**

# Балльно- рейтинговая система

01

# **100 баллов = 60 работа в семестре + 40 проект**

---

## **Баллы в оценку:**

90+ - оценка 5

80+ - оценка 4

60+ - оценка 3

\*проект - обязательно

\*кр - обязательно

\*лабы на одну меньше можно

## **4,5 - автомат + можно повысить на экзамене**

## Работа в семестре: (max 60)

- max 10 баллов – контрольная:

оценка 5: 10 баллов

оценка 4: 8 баллов

оценка 3: 4 балла

оценка 2 – переписывать

- max 50 баллов – лабы:

~ 7 баллов одна лаба

Принцип сдал/не сдал (0 или 7 баллов можно получить)

Сдал = прошли все тесты

В конце семестра могу выборочно поставить половину баллов за лабу – если студент упорно и долго пытается пройти тесты и всё никак не получается победить

## **Проект: (max 40)**

- **На любом языке программирования**
- **Набрать баллы как конструктор**
- **Демонстрировать прогресс по дедлайнам:**

**5 марта – выбор темы, ресерч**

**15 марта – первые шаги**

**5 апреля – прогресс**

**26 апреля – прогресс**

**10 мая – почти готово**

**21,28 мая – очно защита проекта на паре**

## Конструктор проекта (max 40)

Идея – набрать баллы из пула фичей и требований,  
которые в сумме превышают максимальное количество  
баллов

### Обязательные требования max 10 баллов

**1 балл** – Оформленное README проекта (описание, как локально развернуть, картинки/графики/выводы)

**1 балл** – Структурированный проект на уровне директорий (не все файлы в одной папке)

**3 балла** – Документация (подробное описание логики, структуры кода, описание классов/методов), можно как отдельный readme

**3 балла** – применение ООП

**3 балла** – написание тестов

# Конструктор проекта – пулл фичей/требований

- ☐ 5 баллов – Консольный интерфейс (CLI)
  - ☐ 5 баллов – Работа с файлами (чтение/запись)
  - ☐ 5 баллов – Реализация и применения алгоритмов
  - ☐ 5 баллов – Интеграция со сторонним API
  - ☐ 5 баллов – Реализации системы конфигураций (наличие опциональности за счет параметров конфиг)
  - ☐ 10 баллов – Графический интерфейс (GUI), в том числе десктопное приложение
  - ☐ 10 баллов – REST API (минимум 4 endpoint)
  - ☐ 10 баллов – Работа с базой данных
  - ☐ 10 баллов – Интеграции ML модели/ее обучение
  - ☐ 10 баллов – Обоснованное применение шаблона проектирования (например, Singleton, Factory)
  - ☐ 15 баллов – Многопоточность/ассинхронность и тд
- 
- ☐ 2 балла – Типизация в питоне
  - ☐ 2 балла – Логирование (в файл или консоль)
  - ☐ 2 балла – Использование, настройка линтеров
  - ☐ 2 балла – Минимальная настройка CI/CD через Github Actions
  - ☐ 5 балла – Докер образ

# Конструктор проекта – пулл фичей/требований

- [ ] До 10 баллов, если проект на C++
  - [ ] Если с++ 5 баллов – реализация `template` объектов в проекте
  - [ ] Если с++ 10 баллов – использование сторонних фреймворков/библиотек
- 
- [ ] До 10 баллов – ваша фича/фичи
  - [ ] 10 баллов – контрибьют в `opensource`

**Важно:** требования больше технические, нежели сутевые. Но нужно не забывать решить какую-либо проблему/задачу, попытаться сделать качественно и полезно



## Что **нельзя** выбрать в качестве проекта:

- Мобильное приложение
- Только модуль фронтенда
- Телеграмм-бот
- Чисто МЛ-проект/ресерч. МЛ может быть опцией, но сейчас мы про разработку
- Что-то простое, по типу калькулятора ))

## Хочу показать готовый проект

Просто взять готовый нельзя – можно взять его за стартовую точку и по плану работы улучшать проект, например, добавляя новые фичи.

И конечно набирая баллы

# Некоторые идеи

# O2

## Проекты на с++

- Консольное приложение CLI (5)
- Работа с файлами (5)
- Реализация алгоритмов (5)
- Логирование (2)
- Использование template объектов (5)
- Применение с++ (10)

**Работа с изображениями (алгоритмическая обработка, opencv)**

**Вспомогательные утилиты с различным функционалом**

## **Проекты на c++**

- Графический интерфейс (GUI) (10)
- Сторонний фреймворк/библиотека (10)
- Применение c++ (10)

**Менеджер задач**

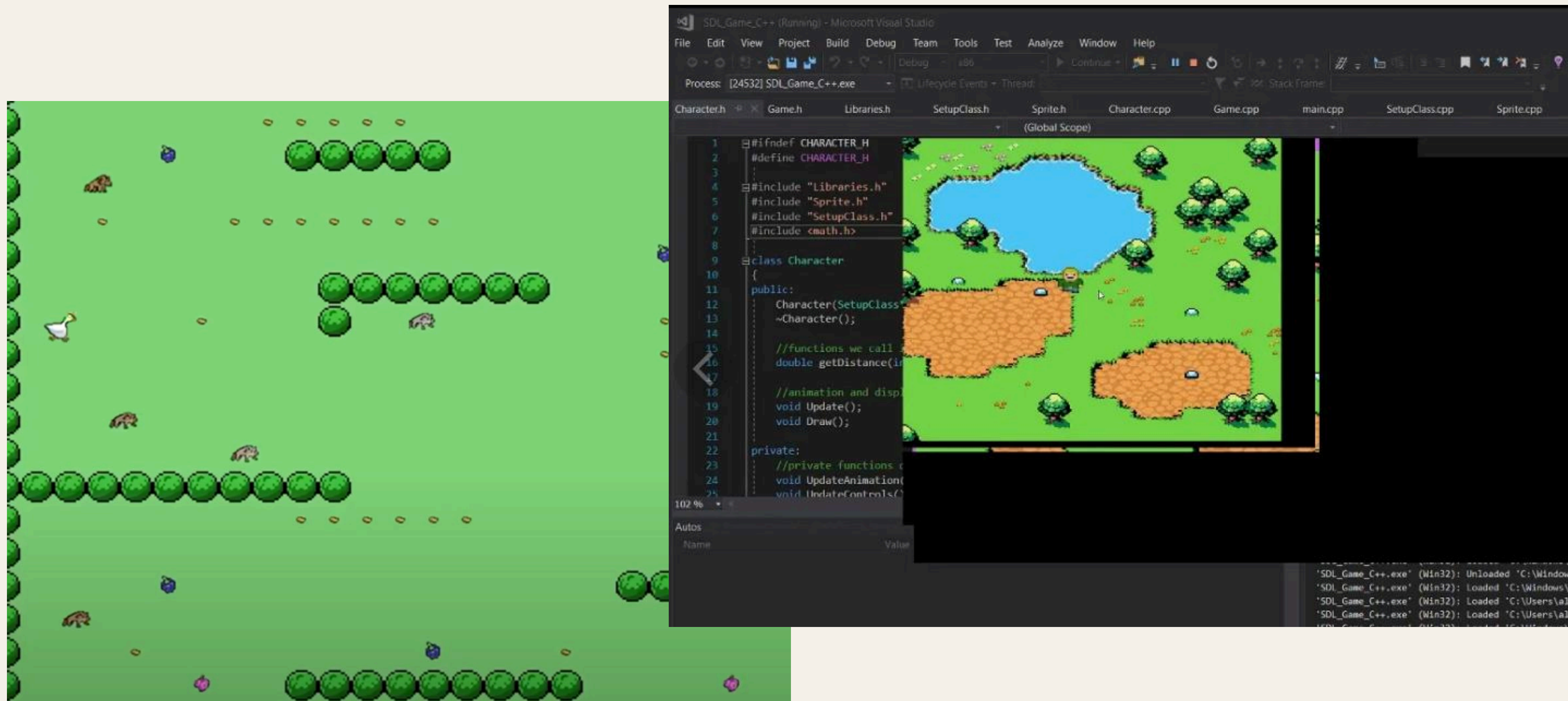
**Визуализация работы чего-то (например нейронок) интерактивная обучалка**

**Десктопная игра**

# Десктопная игра

## Проекты на с++

- интерфейс через какой-нибудь фреймворк (SDL, QT и др)





# llm.c

## Проекты на с++

Опенсорс проект с имплементацией LLM на C от  
Андрея Карпаты

Можно попробовать на с++ имплементировать,  
например, Llama3/Mistral и потестить)

And why I am working on it? Because it's fun. It's also educational, because those 1,000 lines of very simple C are all that is needed, nothing else. It's just a few arrays of numbers and some simple math operations over their elements like + and \*. And it might even turn out to be practically useful with some more work that is ongoing.

<https://twitter.com/karpathy/status/1778153659106533806>

Домашнее  
задание

03

## 5 марта – выбор темы, ресерч

К первому дедлайну:

- Создаете новую ветку
- Создаете папку `final_project`
- Создаете `README_progress.md`

Если сделали раньше и хотите меня пингануть – откройте пулл реквест и добавьте меня в ревьюеры

Внутри `readme`:

- Тема проекта
- Что вы набрали из разбалловки
- Небольшой ресерч темы (похожие проекты/идеи, библиотеки и тд)
- Примерный план работы