ЛЕКЦИЯ I

JAVA OOP: Капсулация

Теодор Костадинов

Съдържание

SOFTWARE ACADEMY
Educational Center Net It

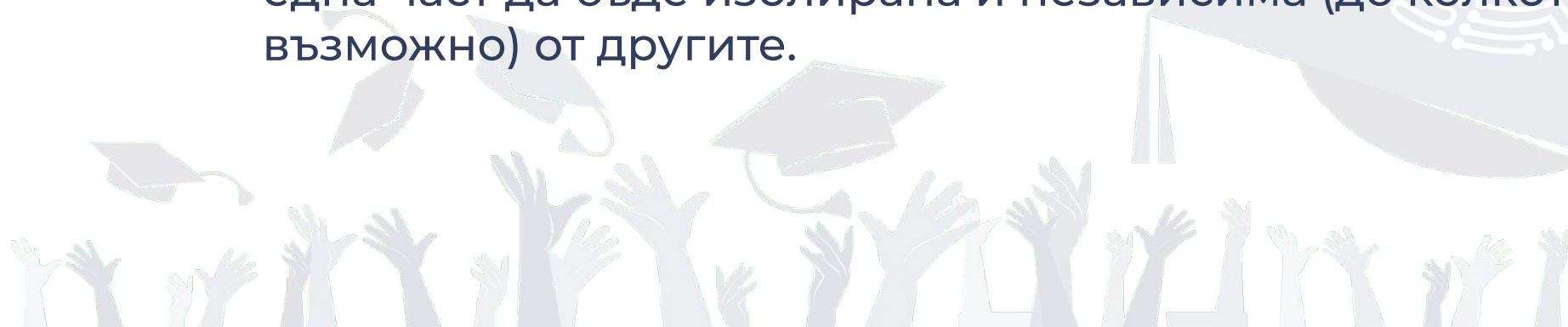
1. Какво и защо?
2. Как се постига?
3. ООП Преговор

Капсулация

Какво е капсулация?

SOFTWARE ACADEMY
Educational Center Net It

- Защита на данните и имплементацията
- Скриването на имплементацията на данните чрез ограничаване на достъпа до мутаторите
- Можем да правим промени на обекта без да се тревожим, че ще счупим другия код, който извиква методите от класа за информация
- Учи ни да разделим отделните части от кода, така че всяка една част да бъде изолирана и независима (до колкото е възможно) от другите.



Какво не е капсулация?

- Когато една програма бива писана програмиста трябва добре да обмисля, кои части от програмата трябва да взаимодействат помежду си и кои не.

Пример за погрешно моделирана комуникация м/у отделни компоненти

- Човек иска на друг човек заем от 5лв. Втория дава на първия портмонето си за да може първия да си вземе 5лв.

Пример за правилно моделирана комуникация м/у отделни компоненти

- Човек иска на друг човек заем от 5лв. Втория вади портмонето си и дава 5лв на първия.

Как се постига?

SOFTWARE ACADEMY
Educational Center Net It

- Пакети
  - Пакетите представляват йерархична структура сходна на директориите в една операционна система. На файловата система пакетите (packages) са репрезентирани като директории. Но в една програма те целят да групират логически отделни компоненти (класове) от програмата.
- Модификатори
  - Всеки един от модификаторите разрешава различна част от програмата да има достъп до различна част от кода.
- Гетъри и сетъри
- Final

ООП
до сега

Какво видяхме до сега?

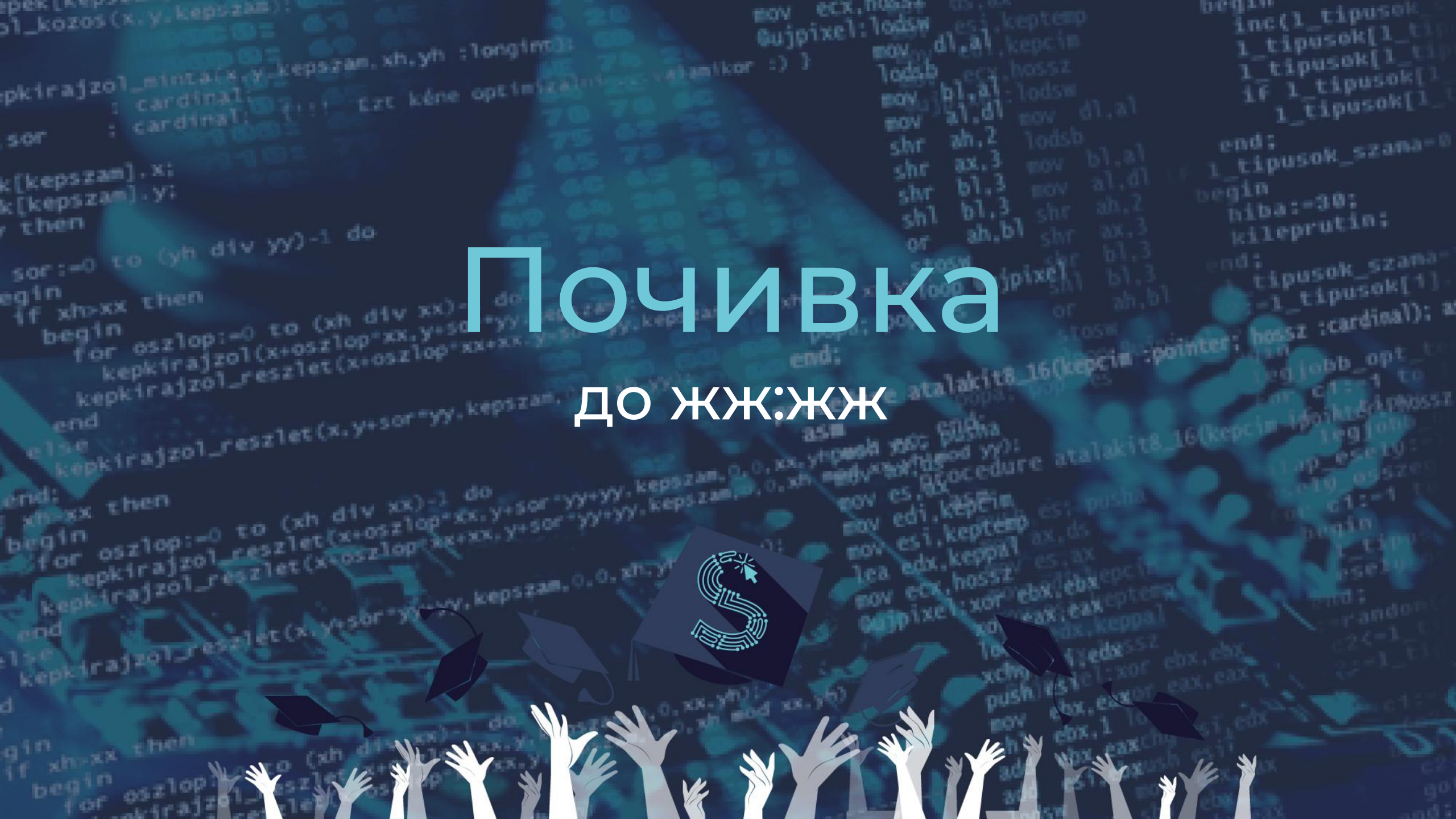
SOFTWARE ACADEMY
Educational Center Net It

- Класове
- Обекти
- Конструктори
- Полета
- Методи
- Статичност
- Енумерация
- Наследяване
- Полиморфизъм
  - Overriding & Overloading
- Абстрактност
  - Interfaces & abstract classes
- Капсулация
  - Getters & setters, modifiers, final

ВЪПРОСИ?

Почивка

до жж:жж



Резюме

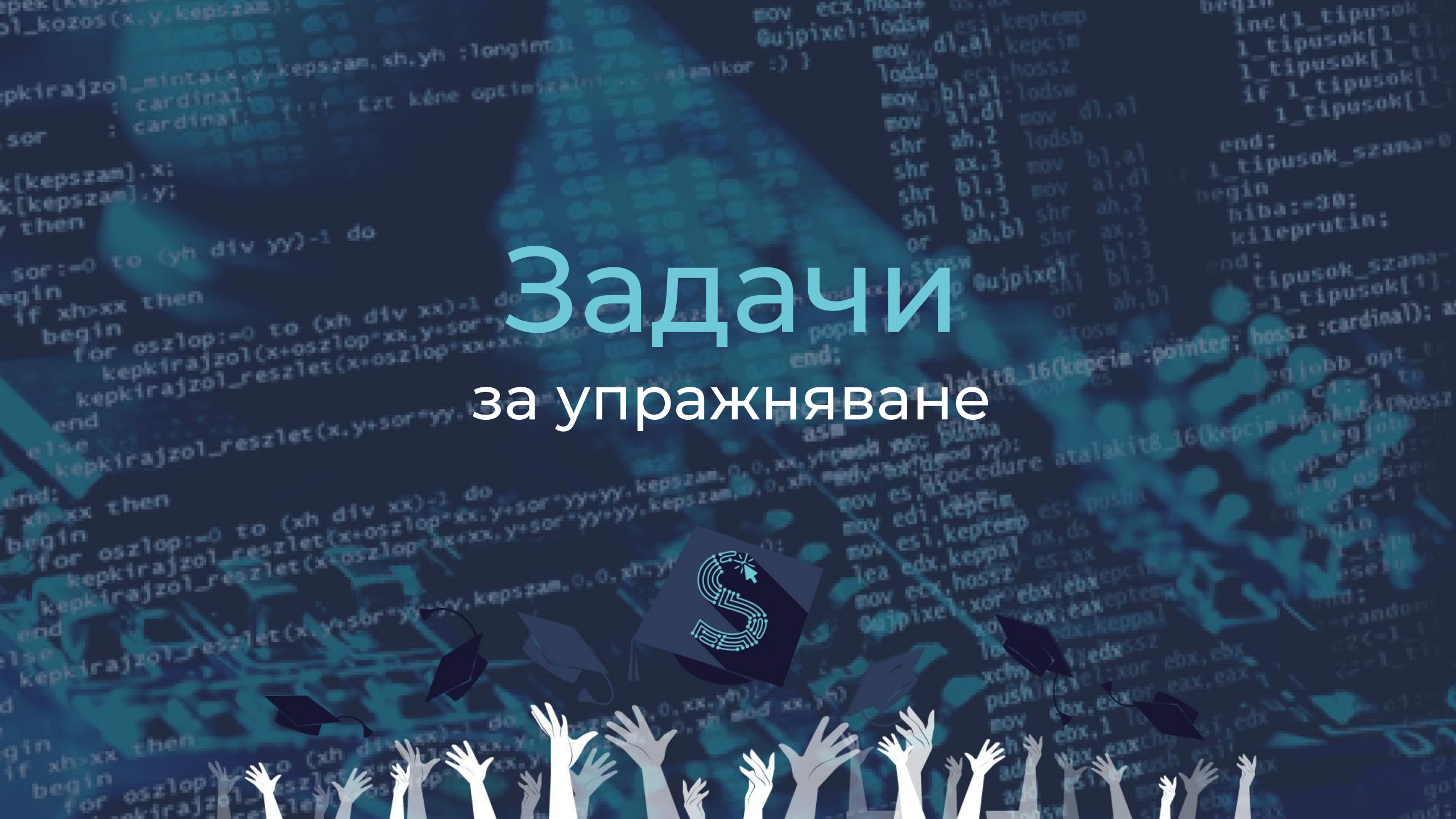
- Капсулация се постига както с гетъри и сетъри и модификатори, така и със самата структура на проекта.

Ресурси

- Docs
- GitHub Repo with Demos
-

Задачи

за упражняване



Това домашно влиза в крайната ви оценка!

Домашно

Качвайте домашното си в ГитХъб и слагайте линка тук:

https://forms.gle/AcvCptCbSDizr2Ay6

Задача 1

Всеки добър ресторант има система за поръчки. Тя служи не само за автоматизация на работата на сервитьорите, но и на кухнята. Сервитьорът трябва да може да създава поръчка за определена маса, а кухнята да сменя статуса й на “готви се”, “приготвена”, съответно сервитьора да може да я смени на “сервирана”. Сервитьорът може да извади сметката за дадената маса, което сменя статуса на “платена” и освобождава масата. Всяка поръчка се състои от ястия и напитки от менюто. Сервитьорът може да въвежда и променя менюто с имена, цени и тип.

Задача 1

Проектът трябва да може:

- Трябва да има две роли: сервитьор и готвач. Те се логват в началото на използването на програмата.
- Готвачът единствено може да вижда новите поръчки и да сменя статуса им на “готви се” и “приготвена”
- Има меню. Сервитьорът може да променя менюто - да добавя или маха ястия. Всяко ястие има име, цена и тип.
- Сервитьорът може да вижда менюто. В него ястията са разделени по тип.
- Сервитьорът може да вижда списък с всички активни поръчки.
- Сервитьорът може да създава и редактира поръчки. Всяка поръчка си има дата и час на създаване и номер на маса. Не може да се създаде повече от една поръчка за маса.

Задача 1

SOFTWARE ACADEMY
Educational Center Net It

Проектът трябва да може:

- Към всяка поръчка може да се добавят или премахват ястия. Всяко ястие може да се добави веднъж или много пъти. Общата цена на поръчката се показва в реално време.
- Сервитьорът може да смени статуса на поръчка на “сервирана”.
- Сервитьорът може да смени статуса на поръчка на “платена”. Тогава му се показва обобщение на поръчката, тя изчезва от списъка с активни поръчки и на тази маса вече може да се прави нова поръчка.
- Има история на поръчките, която сервитьорът може да вижда.

БЛАГОДАРЯ
ЗА ВНИМАНИЕТО!