

СУ „Св. Климент Охридски“, ФМИ

СПЕЦИАЛНОСТ „СОФТУЕРНО ИНЖЕНЕРСТВО“

Обектно-ориентирано програмиране, 2019-2020 г.

Задача за домашно № 1

Онлайн Магазин (1/4)

Проектирайте онлайн магазин подобен на Emag/Amazon.

Реализирайте структурите/класовете на проекта по начина, зададен в изискванията, обсъдени на упражнения

Реализирайте необходимите полета (член-данни) и основни методи за работа с тях, ако такива са необходими. Не се очаква да бъдат реализирани полета, които са от сложен тип, т.е не е необходимо да се свързват отделните структури/класове чрез композиция.

Спазвайте принципите за енкапсулация и качествен код, обсъждани на лекции, упражнения и практикум.

Обектно-ориентирано програмиране, 2019-2020 г.

Задача за домашно № 2

Онлайн Магазин (2/4)

Проектирайте онлайн магазин подобен на Emag/Amazon.

Рефакторирайте кода от домашно 1.

Добавете към класовете подходящи конструктори/деструктори и други методи, които сметнете за необходими.

Реализирайте композиция/агрегация между класовете за да ги свържете смислено. т.е очаква се да реализирате полетата от сложен тип (клас, а не примитивен тип).

Спазвайте принципите за енкапсулация и качествен код, обсъждани на лекции, упражнения и практикум.

Обектно-ориентирано програмиране, 2019-2020 г.

Задача за домашно № 3

Онлайн Магазин (3/4)

Проектирайте онлайн магазин подобен на Emag/Amazon.

Рефакторирайте кода от домашно 2.

Променете класовете, така че да избегнете използването на един и същ код на 2 места чрез използване на шаблони, където е възможно.

Например, ако имате динамичен масив от обект А и друг масив от обект Б, реализирайте темплейтен динамичен масив и го използвайте на мястото на динамичните масиви за обекти от тип А и Б.

Обектно-ориентирано програмиране, 2019-2020 г.

Задача за домашно № 4

Онлайн Магазин (4/4)

Проектирайте онлайн магазин подобен на Emag/Amazon.

Рефакторирайте кода от домашно 3, като използвате **STL** вместо написаните от вас динамични масиви, поинтъри и други където е възможно. Разгледайте класовете ``std::string`` & ``std::vector`` и работата с итератори.

Променете класовете, така че да избегнете използването на един и същ код на 2 места чрез използване на наследяване, където е възможно.