Конспект за Разработка на софтуер

**162 учебни часа** (първи срок: **18 седмици** по **4 часа** + втори срок: **18** седмици по **5** часа)

Идея за съдържание:

* Структура на софтуерен проект
  + Трислоен модел (three-tier architecture)
  + Слой за данни
  + Слой за услуги
  + Слой за UI

Изграждане на практически софтуерен проект

* + Връзка между приложение и база данни, ORM технологии
  + Потребителски интерфейс по избор (Web / desktop / mobile)

Да не се навлиза в технологии, за да се запази фокусът върху структурата на приложението.

## І. Общо представяне на учебната програма

Обучението по предмета има за цел да предостави основната рамка за писане на софтуер и да съедини в себе си знанията от всичко ученото до сега в един голям проект.

## ІІ. Цели на обучението по предмета

Обучението по предмета има за цел да предостави основната рамка за писане на софтуер и да съедини в себе си знанията от всичко ученото до сега в един голям проект.

## ІІІ. Разпределение на учебното време

I срок: 18 седмици x (2 часа теория + 2 часа практика) = 72 часа

II срок: 18 седмици x (2 часа теория + 3 часа практика) = 90 часа

**???**

## ІV. Учебно съдържание

|  |  |  |
| --- | --- | --- |
| **№** | **Наименование на разделите** | **Минимален брой часове** |
| 1 | **Увод в разработката на софтуер - преглед на трислойния модел**  * Слой за данни (data access layer) * Слой за услуги (service layer) * Слой за потребителски интерфейс (UI layer) * Упражнение: създаване на просто трислойно приложение (конзолно, без база данни) | 9 |
| 2 | **Увод в концепцията за тестване - писане на "unit" тестове**  * Концепция за тестване на кода. Концепция за “регресия” и регресионни тестове. * Покриване на кода с тестове. * Упражнение: писане на unit тестове за интеграцията на слоя за услуги със слоя за данни. | 9 |
| 3 | **Увод в концепцията за дебъгване - откриване и отстраняване на проблеми**  * Запознаване с различни методики за търсене на дефекти. * Използване на дебъгер. * Добри практики при дебъгване и отстраняване на дефекти - писане на тестове, които покриват проблема. * Упражнение: дебъгване на код с грешки, отстраняване на грешките и писане на unit тестове. | 9 |
| 4 | **Увод в концепцията за рефакториране и правене на "инкрементални промени"**  * Основни инструменти и похвати за рефакториране на код * Запознаване с идеята за "style guides" и "linters" * Запознаване с идеята за намиране на всички употреби на фрагмент код, навигиране към дефиниция на функция * Добри практики при рефакториране чрез писане на тестове. Концепция за тестваем код * Упражнение: рефакториране на недобре написан код, промяна на кода да стане тестваем, покриване с тестове | 18 |
| 5 | **Увод в инструментите за разработка - IDE, текстови редактори, командни инструменти**  * Разлика между IDE и текстов редактор * Техники за продуктивно използване на IDE - допълнителни разширения и клавишни комбинации * Техники за продуктивно използване на текстов редактор - допълнителни разширения и клавишни комбинации. * Използване на базови инструменти за търсене и промяна на кода. * Упражнение: добавяне и използване на разширения в текстов редактор и в IDE | 4 |
| 6 | **Използване на външни библиотеки. Управление на пакети**  * Търсене, инсталиране и употреба на външни библиотеки, които решават често срещани проблеми. * Четене, ориентация и търсене в документация на външна библиотека. * Запознаване с концепцията за инструменти за управление на пакети (package managers) * Упражнение: инсталиране и използване на външни библиотеки | 9 |
| 7 | **Свързване на приложения с бази от данни**  * Използване на "клиентски библиотеки" за свързване с база от данни * Четене на конфигурационни файлове * Писане на SQL заявки през език за програмиране * Изпълнение на SQL заявки чрез език за програмиране, изпълнение на SQL команди и извличане на резултата от тях * Запознаване с идеята зад "prepared statements" * Създаване на CRUD приложения върху база от данни * Концепция за обектно-релационно съответствие (ORM frameworks) * Упражнения: създаване на CRUD приложение с база от данни с SQL команди (трислойно приложение, конзолно) * Упражнения: създаване на CRUD приложение с база от данни с ORM и (трислойно приложение, конзолно) | 27 |
| 8 | **Създаване на приложения с няколко потребителски интерфейса**  * Разделяне на приложението на три ясно отделени проекта (или библиотеки с класове): за данни, за услуги и за потребителски интерфейс. * Правилно структуриране на проекта: изолиране на бизнес логика, презентационна логика и логика за достъп до данни в съответните проекти (библиотеки с класове) * Създаване на поне два различни потребителски интерфейса за едно и също приложение * Упражнение: изграждане на трислойно приложение с конзолен / GUI / друг презентационен слой | 18 |
| 9 | **Създаване на курсова работа в екип**  * Трислойно приложение със слой за данни, слой за услуги и потребителски интерфейси * Поне два потребителски интерфейса * Използване на база данни * Използване на външни библиотеки * Пълно покритие с unit тестове * Спазване на "style guide" | 36(4 седмици,~ 1 месец) |
| 10 | **Защита на курсова работа**  * Преглед на кода за спазване на добрите практики (code review) * Презентация и защита на проекта | 18 |
|  | **Резерв часове** | 5 |

V. Тематичен план

Като таблицата горе?

## VІ. Очаквани резултати от обучението

Покрива:

* РУ10.5
* РУ10.6
* РУ13.1
* РУ13.2

## VІІ. Авторски колектив

* Радослав Георгиев, Ивайло Бъчваров, <https://hackbulgaria.com>, <https://hacksoft.io>

## VІІІ. Литература

1. Software Development, Design and Coding: With Patterns, Debugging, Unit Testing, and Refactoring 2nd edition, John F. Dooley, Apress, 2017, ISBN 978-1484231524
2. Extreme Programming Explained: Embrace Change, 2nd Edition, Kent Beck, Addison-Wesley, 2005, ISBN 978-0321278654
3. Програмиране 101 към ХакБългария - <https://github.com/HackBulgaria/Programming101-Python-2016>
4. ...