**МИНИСТЕРСТВО НА ОБРАЗОВАНИЕТО И НАУКАТА**

**У Ч Е Б Н А П Р О Г Р А М А**

**ЗА ОТРАСЛОВА ПРОФЕСИОНАЛНА ПОДГОТОВКА**

по

**ОБЕКТНО-ОРИЕНТИРАНО ПРОГРАМИРАНЕ**

**УЧЕБНА ПРАКТИКА ПО: ОБЕКТНО-ОРИЕНТИРАНО ПРОГРАМИРАНЕ**

УТВЪРДЕНА СЪС ЗАПОВЕД № **РД 09 – .........** от......**.**.....**.2017 г.**

**ПРОФЕСИОНАЛНО НАПРАВЛЕНИЕ: 481 „КОМПЮТЪРНИ НАУКИ“**

**ПРОФЕСИИ: 481010 „ПРОГРАМИСТ”**

**481020 „СИСТЕМЕН ПРОГРАМИСТ”**

**481030 „ПРИЛОЖЕН ПРОГРАМИСТ”**

**София, 2017 година**

1. **ОБЩО ПРЕДСТАВЯНЕ НА УЧЕБНАТА ПРОГРАМА**

Учебната програма по „Обектно-ориентирано програмиране”е предназначена за специалности:

* 4810101 „Програмно осигуряване”
* 4810201 „Системно програмиране”
* 4810301 „Приложно програмиране”.

Учебното съдържание в програмата е структурирано в два срока и общо петнадесет раздела, които дават възможност на учениците да получат задълбочени знания и практически умения и компетентности за работа с класове и обекти.

1. **ЦЕЛИ НА ОБУЧЕНИЕТО ПО ПРЕДМЕТА**

Обучението по предмета има за цел учениците да придобият задълбочени знания за обектно-ориентирани конструкции като класове и обекти.

За постигане на основната цел на обучението по „Обектно-ориентирано програмиране”, е необходимо изпълнението на следните под-цели:

* Задълбочаване на знанията и уменията за работа с класове - капсулация, шаблонни класове, наследяване, абстрактни класове и интерфейси, полиморфизъм, управление на изключенията;
* Придобиване на теоретични и практически знания и умения за обектно-ориентирано моделиране и ползване на базови шаблони за дизайн.

1. **УЧЕБНО СЪДЪРЖАНИЕ**

Учебното съдържание е  структурирано в раздели  и теми. За всеки раздел  в програмата е определен минимален брой учебни часове. Учителят разпределя броя учебни часове за нови знания, упражнения и оценяване, при спазване изискванията за минимален брой часове по раздели.

Разликата между броя на учебните часове в учебния план и общия минимален брой, предвиден в учебната програма определя резерва часове. Те се разпределят по теми в началото на учебната година от учителя.

Раздели:

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | Наименование на разделите | Минимален брой часове теория | Минимален брой часове практика |
| 1. | Компонентно тестване (unit testing) | 4 | 4 |
| 2. | Дефиниране на класове за напреднали | 8 | 8 |
| 3. | Шаблонни класове | 4 | 4 |
| 4. | Наследяване, абстракция, интерфейси | 10 | 10 |
| 5. | Полиморфизъм в ООП | 8 | 8 |
| 6. | Работа с обекти | 4 | 4 |
| 7. | Елементи от функционалното програмиране | 4 | 4 |
| 8. | Комуникация между обекти (събития / интерфейси) | 4 | 4 |
| 9. | Изключения (exceptions) | 4 | 4 |
| 10. | Работа с потоци и файлове | 4 | 4 |
| 11. | Базови шаблони за дизайн (design patterns) | 2 | 2 |
| 12. | Разработка на практически проект по ООП | 8 | 8 |
|  | **Общ минимален брой часове** | **64** | **64** |
|  | **Резерв часове** | **8** | **8** |
|  | **Общ брой часове** | **72** | **72** |

1. **ТЕМАТИЧЕН ПЛАН**

###### **Раздел 1. Компонентно тестване (Unit Testing)**

###### Компонентно тестване (Unit Testing) – писане на Unit тестове

Упражнение: писане на Unit тестове (тестване на готов клас)

###### **Раздел 2. Дефиниране на класове за напреднали**

###### Класове, конструктори, деструктори, свойства, методи, събития, статични членове

###### Упражнения: дефиниране на класове (имплементация на клас “множество от числа” по подготвени Unit тестове)

###### Концепция за виртуални среди за изпълнение, компилатори, интерпретатори, среди за изпълнение

###### Памет, стек и хийп, разположение на обектите в паметта. Упражнения за работа с паметта

###### **Раздел 3. Шаблонни класове**

###### Шаблонни класове (Templates / Generics)

###### Упражнения: шаблонни класове (имплементация на шаблонен клас “списък, базиран на масив” по предварително подготвени Unit тестове)

###### **Раздел 4. Наследяване, абстракция, интерфейси**

###### Наследяване и йерархии от класове

###### Упражнения: наследяване на класове и изграждане на йерархия (животни с различни свойства, които се наследяват)

###### Абстракция, абстрактни класове и интерфейси

###### Упражнения: абстракция, абстрактни класове и интерфейси (имплементация на интерфейси + шаблонни класове за структурата от данни “свързан списък”)

###### **Раздел 5. Полиморфизъм в ООП**

###### Полиморфизъм и виртуални методи

###### Упражнения: полиморфизъм и виртуални методи (йерархия от класове за геометрични фигури с виртуални методи за смятане на лице и за отпечатване в текстов вид)

###### Упражнения: изграждане на йерархия от класове, с интерфейси, наследяване и полиморфизъм (йерархия от класове за превозни средства с виртуални методи за смятане на лице и за отпечатване в текстов вид)

###### **Раздел 6. Работа с обекти**

###### Компаратори (сравнение, хеш код), итератори, отражение на типовете (Reflection)

Упражнения: компаратори, итератори, отражение на типовете (имплементация на контейнер-структура “сортиран списък” + итератор + сортиране на студенти по Username)

###### **Раздел 7. Елементи от функционалното програмиране**

###### Променливи от тип “функция”, обратно извикване (Callback), Lambda изрази, библиотеки за функционална обработка на колекции

###### Упражнения: функционална обработка на колекции

###### **Раздел 8. Комуникация между обекти (събития / интерфейси)**

###### Комуникация между обекти в ООП (събития и събитийно програмиране)

###### Упражнения: комуникация между обекти

###### **Раздел 9. Изключения (Exceptions)**

###### Работа с изключения, дефиниране на собствени типове изключения

###### Упражнения: работа с изключения

###### **Раздел 10. Работа с потоци и файлове**

###### Потоци и файлове, текстови и бинарни потоци

###### Упражнения: потоци и файлове

###### **Раздел 11. Базови шаблони за дизайн (Design Patterns)**

###### Запознаване с концепцията за шаблони за обектно-ориентиран дизайн. Шаблони, които вече знаем:

###### Builder: StringBuilder, XmlBuilder

###### Prototype: Object.clone()

###### Composite: структура от данни “дърво”

###### Strategy / Pluggable Behaviour: Filtering with Lambda

###### Visitor / Iterator: итератори в класовете за колекции

###### Adapter: StreamReader

###### Command: Runnable / Action

###### **Раздел 12. Разработка на практически проект по ООП**

###### Разработка на практически проект по ООП (например ООП игра или друга система, структурирана с класове и обекти) – моделиране на йерархията от класове

###### Практическа работа по ООП проекта – имплементация

###### Практическа работа по ООП проекта – имплементация

###### Консултации по ООП проекта – довършителни задачи

1. **ОЧАКВАНИ РЕЗУЛТАТИ ОТ ОБУЧЕНИЕТО**

В края на обучението по учебния предмет, учениците придобиват компетентности за:

* Работа по задачи в обектно-ориентирания модел;
* Използване на класове, абстрактни класове и интерфейси, капсулация, наследяване, полиморфизъм и управления на изключения;
* Реализиране на комуникация между обекти в обектно-ориентирана среда;
* Моделиране и създаване на дизайн на софтуер. Обектно-ориентирано моделиране на софтуерни системи и използване на базови шаблони за дизайн;

1. **АВТОРСКИ КОЛЕКТИВ**

Програмата е разработена от:

* д-р Никола Вълчанов, Програмиста, ФМИ към ПУ, Пловдив
* д-р Светлин Наков, СофтУни, София

Програмата е обсъдена, коригирана и оформена от експертна група към Национална програма „Обучение за ИТ кариера“ към МОН с представители на БАСКОМ, БАИТ, ИКТ клъстер и Българска аутсорсинг асоциация в състав:

* д-р Стела Стефанова, ТУЕС към ТУ, София
* Любомир Чорбаджиев, ТУЕС към ТУ, София
* Радослав Георгиев, HackSoft, HackBulgaria, София
* Веселина Карапеева, ОМГ "Акад. К. Попов", Пловдив
* Ангел Георгиев, СофтУни, София
* Ивайло Бъчваров, HackSoft, HackBulgaria, София
* Мирослав Миронов, Мусала Софт, София
* Владимир Начев, EPAM, София
* Димитър Димитров, БАИТ, София

1. **ЛИТЕРАТУРА**
2. Наков С., Колев В. и колектив, Въведение в със C#, София, 2015, ISBN 978-954-400-527-6, <http://www.introprogramming.info/intro-csharp-book/>
3. Наков С. и колектив, Въведение в с Java, София, 2008, ISBN 978-954-400-055-4, <http://www.introprogramming.info/intro-java-book/>
4. McLaughlin, B. D.; Pollice, G. & West, D. (2007), Head first object-oriented analysis and design - a brain-friendly guide to OOA&D., O'Reilly
5. Gamma, e. a. (1994), Design Patterns: Elements of Reusable Object-Oriented Software, Addison Wesley Professional
6. Weisfeld, M. (2013), The Object-Oriented Thought Process (Developer's Library) 4th Edition, Addison-Wesley Professional

## Министерство на образованието и науката (МОН)

* Настоящият курс (презентации, примери, задачи, упражнения и др.) е разработен за нуждите на Национална програма "**Обучение за ИТ кариера**" на МОН за подготовка по професия "Приложен програмист".



* Курсът е базиран на учебно съдържание и методика, предоставени от **фондация "Софтуерен университет"** и се разпространява под **свободен** **лиценз CC-BY-NC-SA** (Creative Commons Attribution-Non-Commercial-Share-Alike 4.0 International).

