

## KATALOG ZNANJA

### 1. IME PREDMETA

#### NAPREDNO PROGRAMIRANJE

### 2. SPLOŠNI CILJI

#### Splošni cilji predmeta so:

- naučiti naprednih programskih jezikov za oblikovanje učinkovite programske opreme;
- predstaviti delovanje vzorcev oblikovanja programske opreme ter postopek reševanja zastavljenih problemov pri objektno orientiranem programiranju;
- usposobiti za razreševanje težav, kritično mišljenje, analitično razmišljanje, sprejemanje odločitev;
- razviti digitalno in računsko pismenost ter kreativnost;
- naučiti načrtovanja in določanja prioritet, samoupravljanja, samostojnega učenja ter samorefleksije.

#### Specifično strokovno usmerjeni cilji so:

- pisati objektno orientirane programske kode;
- oblikovati strukturirane kode v poenotenem jeziku na modeliranje;
- poznati prednosti uporabe naprednih programskih jezikov (npr. učinkovitost delovanja, zniževanje stroškov, izboljšana berljivost ter razširljivost programske kode, i. d.).

### 3. PREDMETNO SPECIFIČNE KOMPETENCE

#### Pri predmetu študenti poleg generičnih pridobijo naslednje kompetence:

1. analiziranje ključnih komponent objektno orientiranega programiranja ter različnih načrtovalskih vzorcev zasnove programske opreme;
2. oblikovanje različnih diagramov razredov s pomočjo poenotenega jezika za modeliranje (UML);
3. implementiranje programskih rešitev z uporabo načrtovalskih vzorcev;
4. raziskovanje praktične uporabe načrtovalskih vzorcev.

## 4. OPERATIVNI CILJI

INFORMATIVNI CILJI	FORMATIVNI CILJI
<b>1. Analiziranje ključnih komponent objektno orientiranega programiranja ter različnih načrtovalskih vzorcev zasnove programske opreme</b>	
<ul style="list-style-type: none"><li>• opiše karakteristike objektno orientirane paradigme ter različne vrste povezav med razredi;</li><li>• definira značilnosti paradigem objektno orientiranega programiranja (npr. inkapsulacija, polimorfizem, konstruktorji ter destruktorji, uporabniški vmesniki, metoda redefiniranja, predloge, vsebniki, i. d.);</li><li>• analizira povezave med objektno orientiranimi razredi (npr. dedovanje, realizacija, odvisnost, agregacija kompozicija, i. d.);</li><li>• ovrednoti razmerje med objektno orientirano paradigmo ter vzorci oblikovanja programske opreme.</li></ul>	<ul style="list-style-type: none"><li>• izpelje vzorce (design patterns) iz primerov oblikovanja programske opreme iz različnih vrst vzorcev.</li></ul>
<b>2. Oblikovanje različnih diagramov razredov s pomočjo poenotenega jezika za modeliranje (UML)</b>	
<ul style="list-style-type: none"><li>• definira razrede UML in način pridobivanja razrednih diagramov in programske kode z uporabo orodja UML;</li><li>• analizira način pridobivanja razrednih diagramov in programske kode z uporabo orodja UML;</li><li>• ovrednoti postopek pregledovanja programske kodo ter določi ustrezno orodje UML za oblikovanje razrednih diagramov.</li></ul>	<ul style="list-style-type: none"><li>• načrtuje in izdela razredne diagrame z uporabo orodij UML.</li></ul>
<b>3. Implementiranje programskih rešitev z uporabo načrtovalskih vzorcev</b>	
<ul style="list-style-type: none"><li>• definira postopek implementacije vzorcev oblikovanja programske opreme;</li><li>• analizira postopek uporabe ustreznega programskega jezika ter integriranega razvojnega okolja za pisanje programske kode;</li></ul>	<ul style="list-style-type: none"><li>• izdela aplikacijo, ki je izpeljana iz razrednih diagramov UML;</li><li>• napiše programsko kodo, ki implementira vzorec oblikovanja programske opreme za določen namen.</li><li>• izdela programsko kodo, ki vsebuje vzorce oblikovanja programske opreme ter ima funkcijo pisanja varne kode;</li></ul>

<ul style="list-style-type: none"><li>• ovrednoti postopek uporabe vzorcev oblikovanja programske opreme na podlagi praktičnih primerov.</li></ul>	
<b>4. Raziskovanje praktične uporabe načrtovalskih vzorcev</b>	
<ul style="list-style-type: none"><li>• definira razpon uporabe vzorcev oblikovanja programske opreme, s pomočjo ustreznih primerov vrst vzorcev;</li><li>• analizira ter povezuje vzorce oblikovanja programske opreme;</li><li>• ovrednoti ustrezno rabo vzorcev oblikovanja programske opreme z nizom vnaprej določenih scenarijev.</li></ul>	<ul style="list-style-type: none"><li>• izdelava predstavitev, v kateri kritično osvetli ter zagovarja različne izbrane vzorce oblikovanja programske opreme za vnaprej določene scenarije.</li></ul>

## **5. OBVEZNOSTI ŠTUDENTOV IN POSEBNOSTI V IZVEDBI**

Število kontaktnih ur: 80 ur (40 ur predavanj, 40 ur laboratorijskih vaj).

Število ur samostojnega dela študenta: 100 (študij literature, delo z besedilom, delo s programsko opremo, priprava pisnih besedil in predstavitve ...).