

## KATALOG ZNANJA

### 1. IME PREDMETA

#### PROGRAMIRANJE

### 2. SPLOŠNI CILJI

#### Splošni cilji predmeta so:

- predstaviti temeljne koncepte različnih vrst programiranja;
- predstaviti osnovne algoritme in njihovo uporabnost za programiranje;
- predstaviti značilnosti programskih paradigem;
- razviti medosebne veščine ter kulturno zavest in izražanje;
- razvijati sposobnost načrtovanja in določanja prioritet, samoupravljanja ter samorefleksije;
- razviti kritično mišljenje, analitično razmišljanje ter digitalno pismenost;
- razviti aktivni pristop pri iskanju virov informacij in znanja;
- naučiti za načrtovanje projektne dela.

#### Specifično strokovno usmerjeni cilji so:

- obvladati postopek programiranja aplikacij s pomočjo integriranega razvojnega okolja (IDE);
- razumeti pomen algoritmov in njihovo uporabo pri pisanju izvorne kode za aplikacijo;
- reševanti zastavljene izzive pri programiranju v integriranem razvojnem okolju;
- implementirati algoritme v izbrani programski jezik, znotraj integriranega razvojnega okolja;
- uporabiti postopkovno, objektno ter dogodkovno programiranje.

### 3. PREDMETNO SPECIFIČNE KOMPETENCE

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

1. definiranje algoritmov za reševanje izzivov in strukturiranje procesa razvoja programskih rešitev;
2. definiranje značilnosti postopkovnega, objektnega in dogodkovnega programiranja;
3. implementacija algoritmov z uporabo integriranega razvojnega okolja;
4. razhroščevanje in odpravljanje napak ter zagovarjanje pomembnosti standardov programskih jezikov.

## 4. OPERATIVNI CILJI

INFORMATIVNI CILJI	FORMATIVNI CILJI
<b>1. Definiranje algoritmov za reševanje izzivov in strukturiranje procesa razvoja programskih rešitev</b>	
<ul style="list-style-type: none"><li>● spozna definicijo algoritma in postopek izdelave aplikacije;</li><li>● razloži razmerje med algoritmi in kodo;</li><li>● razčleni korake izdelave od pisanja programske kode do zagona;</li><li>● razčleni strukture za razvoj vmesnikov (JSON, XML, HTML, CSS, i.d.)</li><li>● utemelji postopek generiranja kode, vloge predprocesorja, prevajalnika in povezovalca in tolmača.</li></ul>	<ul style="list-style-type: none"><li>● izdelava algoritme za izvedbo določene operacije;</li><li>● izvede implementacijo algoritma v ustrezni programski jezik in določi razmerje med napisanim algoritmom in različico programske kode.</li></ul>
<b>2. Definiranje značilnosti postopkovnega, objektnega in dogodkovnega programiranja.</b>	
<ul style="list-style-type: none"><li>● spozna ključne komponente integriranega razvojnega okolja,</li><li>● spozna nadzorne strukture ter način posredovanja parametrov, razredov in dogodkov utemelji značilnosti podatkovnih tipov ter vlogo konstant in spremenljivk;</li><li>● načrtuje vhodne in izhodne metode pisanja izvorne kode;</li><li>● ovrednoti paradigme postopkovnega, objektnega in dogodkovnega programiranja, njihove značilnosti in povezave med njimi.</li></ul>	<ul style="list-style-type: none"><li>● izdelava analizo značilnosti in strukture izvorne kode določene aplikacije, ki izvaja postopkovne, objektne in dogodkovne paradigme,</li><li>● izdelava kategorizacijo vsake komponente integriranega razvojnega okolja,</li><li>● izdelava razširitve z uporabo urejevalnikov besedila za urejanje kode (npr. Notepad++, Atom, Sublime Text, i. d.)</li></ul>
<b>3. Implementacija algoritmov z uporabo integriranega razvojnega okolja.</b>	
<ul style="list-style-type: none"><li>● spozna postopek implementacije integriranega razvojnega procesa za aplikacijo,</li><li>● analizira izzive, povezane z varnostjo izvorne kode, ter postopek odpravljanja težav,</li><li>● utemelji razliko med uporabo integriranega razvojnega okolja za razvoj aplikacij in razvoj brez IDE.</li></ul>	<ul style="list-style-type: none"><li>● izdelava enostavno aplikacijo, ki izvaja algoritme v določenem programskem jeziku znotraj integriranega razvojnega okolja,</li><li>● izvede varnostni pregled programske kode,</li></ul>

#### **4. Razhroščevanje in odpravljanje napak ter zagovarjanje pomembnosti standardov programskih jezikov**

<ul style="list-style-type: none"><li>● spozna specifične pisanje dokumentacije za odpravljanje napak,</li><li>● razišče napredke pri razvoju programske opreme na izbranem področju (agilni razvoj, scrum),</li><li>● razčleni standarde programiranja in njihove prednosti njihove uporabne za pisanje izvorne kode,</li><li>● spozna postopek odpravljanja napak in možnosti za odpravljanje napak znotraj integriranega razvojnega okolja.</li><li>● kombinira možnosti za razvoj varnejših in robustnejših aplikacij.</li></ul>	<ul style="list-style-type: none"><li>● izvede načrtovan postopek odpravljanja napak v kodi,</li><li>● odpravi ranljivosti izvorne kode znotraj integriranega razvojnega okolja</li><li>● izdelava končno projektno dokumentacijo za določeno aplikacijo.</li><li>● zagovarja standarde programiranja in njihov pomen za delo v skupini kot tudi za individualno delo.</li></ul>
--	--

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

Število kontaktnih ur: 70 ur (25 ur predavanj, 45 ur laboratorijskih vaj).

Število ur samostojnega dela študenta: 150 (študij literature, delo z besedilom, študij primerov, priprava pisnih izdelkov, izdelava aplikacije, predstavitev ...)