

KATALOG ZNANJA

1. IME PREDMETA

PROGRAMIRANJE 1

2. SPLOŠNI CILJI PREDMETA

Splošni cilji predmeta so:

- razviti odgovornosti za načrtno in organizirano delovanje;
- pridobiti zavedanje o pomembnosti in kompleksnosti programskih jezikov;
- spoznati pomen zavestnega spremljanja novosti na področju programiranja;
- spoznati vplive programiranja na splošne družbene vrednote;
- razviti sposobnost razvoja lastnega namenskega programa.

Specifično strokovno usmerjeni cilji so:

- razumeti in uporabljati koncepte programiranja;
- razumeti programske jezike, ki se uporabljajo za pisanje programov;
- pridobiti sposobnost samostojnega dela pri vzpostavitvi enostavnega programa;
- pridobiti zmožnost logičnega razmišljanja;
- razumeti pomen načrtovanja programov in izvajati pisanje programov;
- razumeti pomen testiranja programov;
- razumeti dekompozicijo problema na več manjših in lažje obvladljivih.

3. PREDMETNOSPECIFIČNE KOMPETENCE

Pri predmetu si študenti poleg generičnih pridobijo naslednje kompetence:

1. poznavanje osnov programiranja;
2. izvajanje programiranja;
3. izvajanje objektno orientiranega programiranja;
4. razumevanje definiranja in obravnavanja izjem;
5. načrtovanje in razvoj programske opreme.

4. OPERATIVNI CILJI

INFORMATIVNI CILJI	FORMATIVNI CILJI
1. Poznavanje osnov programiranja:	
<ul style="list-style-type: none">• opiše faze življenjskega cikla programske opreme;• razloži vlogo programskih jezikov in nekatere delitve programskih jezikov;• našteje potrebno programsko opremo za	<ul style="list-style-type: none">• izvede predhodno raziskovanje in analizo problema, ki ga rešuje;• izvede namestitve prevajalnika in razvojnega okolja;• izvede prevod in zagon napisanega

<p>razvoj aplikacij;</p> <ul style="list-style-type: none"> • pojasni avtorske pravice pri pisanju in uporabi programske opreme; • razume razvojno okolje in programski jezik C#; • razloži pomembnost spremljanja razvoja programske opreme, orodij in storitev, ki se uporabljajo pri tem (npr. agilni razvoj); • razume standarde za izdelavo programske opreme za uporabnike s posebnimi potrebami (ang. Web Accessibility Initiative). 	<p>programa;</p> <ul style="list-style-type: none"> • izvede prenos prevedenega programa na drug sistem z istim ali drugim operacijskim sistemom; • določi potrebne licence za razvojno okolje, ki ga uporablja za razvoj programske opreme; • uporabi razvojno okolje za pisanje programske opreme; • izvede prenos napisanega in prevedenega programa v razvojnem okolju med različnimi sistemi; • uporabi strokovno literaturo in portale, ki so namenjeni širjenju znanja in dela na odprtokodnih rešitvah in orodjih (npr. Stack Overflow ...); • uvede nove pristope in tehnologije pri razvoju programske opreme.
<p>2. Izvajanje programiranja:</p>	
<ul style="list-style-type: none"> • razume zgradbo programa; • opredeli spremenljivke in konstante; • razlikuje podatkovne tipe in si zamisli najprimernejšega; • loči aritmetične in logične operacije; • razume krmilne stavke; • razloži, kako se problem strukturira na manjše probleme in manjše probleme rešuje z uporabo metod; • pojasni metode z različnimi podpisi; • loči med klicem parametrov metode po vrednosti in po referenci; • opredeli eno in večdimenzionalna polja. 	<ul style="list-style-type: none"> • izdelava osnovne programe v izbranem programskem jeziku; • v osnovnih programih določi in odpravi sintaktične napake; • uporabi enostavne podatkovne tipe; • določi napake, ki se pojavijo zaradi predstavitve števil v računalniku in jih ustrezno obrazloži; • uporabi osnovne aritmetične in logične operacije; • v programu uporabi pogojne in brezpogojne vejitve; • pripravi zapis v zanki ponavljajočih se delov programa; • izvede delitev daljšega programa na smiselne enote; • izdelava program v vsaj enem splošno namenskem programskem jeziku (Java, Python, javascript, C++, C# ...).
<p>3. Izvajanje objektno orientiranega programiranja:</p>	
<ul style="list-style-type: none"> • zamisli si, kako napisati razred kot predlogo objekta; • definira razred ter mu opredeli lastnosti in metode; • primerja razredne spremenljivke in spremenljivke izvoda; • razloži kontrolo dostopa do podatkov v razredu; • razloži delovanje znotraj nastalih 	<ul style="list-style-type: none"> • izdelava zapis preprostih razredov, iz razredov izpelje izvode, lastnosti in obnašanje konkretnega objekta abstrahira v razred; • uporabi konstruktorje razredov; • izdelava zapis destruktora; • izvede naloge, da spremenljivke izvoda in razreda skrije v definiciji razreda, dostop do njih omogoča prek javnih

<ul style="list-style-type: none">objektov;razume abstraktne razrede in vmesnike;razlikuje med virtualnimi in ne-virtualnimi metodami;razume polimorfizem.	<ul style="list-style-type: none">metod;izdela načrt za hierarhijo razredov in iz načrta izdela programsko kodo;predstavi razlike med abstraktnim razredom in vmesnikom in samostojno pridobi izkušnje uporabe enega ali drugega.
4. Razumevanje definiranja in obravnavanja izjem:	
<ul style="list-style-type: none">pojasni razliko med sintaktičnimi in semantičnimi napakami v programu, ki se pojavijo med prevajanjem, med izvajanjem in logičnimi napakami;razloži odpravljanje napak v sintaksi;razloži, kako pravilno obravnavati izjeme;razume, kako poiskati logične napake z razhroščevalnikom.	<ul style="list-style-type: none">izvede odpravo sintaktične napake v programu glede na sporočila prevajalnika;izvede test za svoj delujoči program;izvede test za poljuben program in izdela poročilo o izjemah in napakah, ki jih je v programu odkril;izvede dodajanje obravnave izjem napisanemu programu;uporabi razrede izjem razvojnega okolja in določi lastne razrede izjem;uporabi orodja za razvoj programske opreme (razhroščevalnik ...).
5. Načrtovanje in razvoj programske opreme:	
<ul style="list-style-type: none">opredeli ustrezne informacije, ki jih potrebuje za izdelavo oz. posodobitev programskih rešitev;zamisli si razvoj programske opreme na različnih platformah (namizne, spletne, mobilne, vgradne ...);razloži specifikacije, načrtuje usklajevanje z možnostmi tehničnih rešitev in si zamisli spremembe ter utemelji predloge izboljšav.	<ul style="list-style-type: none">samostojno izdela načrt za arhitekturo manj zahtevne programske rešitve;izbere ustrezno tehnologijo glede na dani primer (programski jezik, podatkovno bazo, storitve, strojno opremo ...);izdela programske module oziroma aplikacije v izbranem programskem jeziku;izdela povezavo programskih rešitev ter neodvisnih storitev in tehnologij (oblačne storitve, moduli za varovanje podatkov, zaščita osebnih podatkov ...).

5. OBVEZNOSTI ŠTUDENTOV IN POSEBNOSTI V IZVEDBI

Število kontaktnih ur: 70 (35 ur predavanj, 35 ur laboratorijskih vaj).

Število ur samostojnega dela študenta: 150 (študij literature, delo z besedilom, študij primerov, reševanje in analiza nestandardiziranih vprašalnikov in anket, priprava pisnih besedil, predstavitev, izvedba javnega nastopa, igre vlog ...).