

РЕПУБЛИКА МАКЕДОНИЈА  
 МИНИСТЕРСТВО ЗА ОБРАЗОВАНИЕ И НАУКА  
 Биро за развој на образованието  
 СКОПЈЕ 2

Примено:		15.09.2014	
Орг единица	Без	Примок	Вредност
07	12/113		



Врз основа на член 55 став 1 од Законот за организација и работа на органите на државната управа („Службен весник на Република Македонија” бр. 58/00, 44/02 и 82/08, 167/10 и 51/11) и согласно член 22 став 1 од Законот за средно образование („Службен весник на Република Македонија” бр. 44/95, 24/96, 34/96, 35/97, 82/99, 29/02, 40/03, 42/03, 67/04, 55/05, 113/05, 35/06, 30/07, 49/07, 81/08, 92/08, 33/10, 116/10, 156/10, 18/11, 51/11, 6/12, 100/2012, 24/2013 и 41/2014) министерот за образование и наука ја донесе Наставната програма по програмски јазици, избран предмет во природно-математичко подрачје А, четврта година на гимназиското образование.



# НАСТАВНА ПРОГРАМА

МИНИСТЕРСТВО ЗА ОБРАЗОВАНИЕ И НАУКА  
БИРО ЗА РАЗВОЈ НА ОБРАЗОВАНИЕТО

## ПРОГРАМСКИ ЈАЗИЦИ

IV ГОДИНА



Скопје, август 2014

ГИМНАЗИСКО ОБРАЗОВАНИЕ

## 1. ВОВЕД

Наставата по предметот **програмски јазици** на учениците им овозможува проширување на претходно стекнатите знаења и здобивање вештини од областа на информатиката, како и можност за креативно користење на компјутерот во наставата и учењето со примена на алгоритми и програмирање во C++.

Според наставниот план за гимназиско образование во природно - математичкото подрачје – комбинација А, предметот програмски јазици во четврта година има статус на изборен предмет и се изучува три часа неделно, односно 99 часа годишно.

Учениците со овој наставен предмет ќе развиваат способности за успешно користење на интегрирани околии за програмирање при изготвување на едноставни компјутерски програми што ќе им овозможи успешно да продолжат со понатамошно изучување на програмирањето. Оспособеноста на учениците за програмирање ќе им овозможи и олеснување во процесот на учење, како и нивно вклучување во процесот на новите предизвици на животот.

## 2. ОПШТИ ЦЕЛИ

Ученикот/ученичката:

- да усвојува поими, правила, постапки и процедури за изработка на алгоритми и програми со користење на даден програмски јазик;
- да знае да ги применува поимите, правилата, постапките и процедурите кои се користат во програмирањето и програмските јазици;
- да се оспособи правилно да користи интегрирани околии за програмирање;
- да се оспособи да изработува програми со одреден програмски јазик;
- да се запознае со концептот на рекурзија и покажувачи, со постапката на пребарување, сортирање и структурирање на податоци, стандардната библиотека на шаблони и настан-базирано програмирање во програмскиот јазик C++;
- да усвојува и применува дополнителни специфики на програмскиот јазик C++;
- да се оспособи за изработка на едноставни програми со користење на: рекурзија, покажувачи, линеарно и бинарно пребарување, функции-членки, податоци од тип структура, стандардната библиотека на шаблони и визуелни настани во програмскиот јазик C++;
- да се оспособува за комбинирање различни техники при изработка на готови програми;
- да се оспособи за изготвување на програми со едукативна, креативна и/или забавна содржина;
- да усвојува поими, правила, постапки и процедури за изработка на проекти со програмирање;
- да усвојува поими, правила, постапки и процедури за изготвување на продукти со програмирање;
- да се оспособува да истражува и да ја оценува точноста, релевантноста, соодветноста и можностите на електронските информациски извори;
- да го проширува и продлабочува стекнатото информатичко знаење низ практично реализирање на програми со изучените техники во програмскиот јазик C++;
- да се оспособи да ја планира и организира својата работа на конкретен алгоритамски проблем;
- да се оспособи да комбинира елементи во нова целина;
- да се оспособи да изготви целосен проект.



### 3. КОНКРЕТНИ ЦЕЛИ

Програмско подрачје 1: Функциски шаблони, рекурзија и рекурзивни функции (10 часа)			
Цели	Содржини	Поими	Активности и методи
<p>Ученикот/ученичката:</p> <ul style="list-style-type: none"> <li>- да се потсети за примена на функции и обременување на функции;</li> <li>- да знае кога може да користи функциски шаблон;</li> <li>- да се оспособи да дефинира функциски шаблон;</li> <li>- да се оспособи да препознава и реши проблем со функциски шаблон;</li> <li>- да го сфати и да го објасни поимот за рекурзивна функција;</li> <li>- да ја сфати функцијата на основниот чекор во рекурзијата;</li> <li>- да ја сфати функцијата на рекурзивниот чекор во рекурзијата;</li> <li>- да сфати кои се условите кои треба да важат за да терминира рекурзивната функција;</li> <li>- да се оспособи за правилно користење на рекурзивни функции;</li> <li>- да се оспособи за решавање на проблеми со рекурзивни функции</li> <li>- да ги сфати сличностите и разликите меѓу итерација и рекурзија;</li> <li>- да ја сфати мемориската</li> </ul>	<p>Потсетување за примена на функции, функции од математичка библиотека, функции со различен број на параметри</p> <p>Потсетување за пренесување на вредност преку вредносен и референцен параметар</p> <p>Потсетување за обременување на функции</p> <p>Функциски шаблони</p> <p>Рекурзија и рекурзивни функции</p> <p>Примери за користење на рекурзија</p> <p>Рекурзија наспроти</p>	<p>Шаблон (темплејт)</p> <p>Шаблонска параметарска листа</p> <p>Формален типизиран параметар</p> <p>Рекурзија</p> <p>Рекурзивна функција</p> <p>Основен чекор или терминален чекор</p> <p>Рекурзивен повик или рекурзивен чекор</p>	<ul style="list-style-type: none"> <li>- Потсетување на претходно изучените концепти за функции преку изработка на соодветни програми.</li> <li>- Дискусија на концептот за функциски шаблони и нивна поврзаност со обременети функции.</li> <li>- Дискусија за случаите во кои можат да се применуваат функциски шаблони.</li> <li>- Демонстрација на примери за функциски шаблон (пример: функциски шаблон за максимум).</li> <li>- Дискусија за дефинирање на функциски шаблони со посебен осврт на специфицирањето на податочните типови во шаблонот.</li> <li>- Дискусија за инстанците во фамилијата на функции кои се дефинираат со даден функциски шаблон преку примери (на пример да се разгледаат сите функции кои се дефинирани со шаблон за максимум ако податочните типови можат да бидат int, double или float).</li> <li>- Објаснување на концептот на рекурзија.</li> <li>- Дефинирање на рекурзивни функции.</li> <li>- Разгледување на некои едноставни рекурзивни функции (пресметка на факториел, степен, фибоначиев број...).</li> <li>- Дискусија за основниот случај во рекурзивна функција.</li> <li>- Дискусија за разделување на проблемот во два концептуални делови – дел кој програмата знае да го реши и дел кој не знае како да го реши.</li> <li>- Дискусија за правилата кои треба да важат за да терминира една рекурзивна функција.</li> </ul>



сложеност на рекурзивните функции.	итерација		<ul style="list-style-type: none"> <li>- Изработка на рекурзивни функции.</li> <li>- Изработка на програми кои искористуваат рекурзивни функции.</li> <li>- Дискусија за сличностите и разликите на итерацијата и рекурзијата.</li> <li>- Дискусија за мемориската сложеност на рекурзивните функции.</li> </ul>
------------------------------------	-----------	--	--

**Програмско подрачје 2: Алгоритми кај (повеќе-димензионални) низи и графови (22 часа)**

<i>Цели</i>	<i>Содржини</i>	<i>Поими</i>	<i>Активности и методи</i>
<p>Ученикот/ученичката:</p> <ul style="list-style-type: none"> <li>- да умее да распознае проблеми од секојдневниот живот кои вклучуваат пребарување низ податоци</li> <li>- да ја сфати работата на алгоритмот за бинарно пребарување</li> <li>- да ги знае условите кои треба да ги задоволува низата за да може да се искористи алгоритмот за бинарно пребарување</li> <li>- да ја разбира разликата меѓу линеарно и бинарно пребарување</li> <li>- да умее да ги имплементира алгоритмите за линеарно и бинарно пребарување</li> <li>- да ја сфати работата на алгоритмите за сортирање (сортирање со селекција, кофа – сортирање и сортирање со спојување)</li> <li>- да умее да распознае проблеми во кои се користат алгоритмите за сортирање;</li> </ul>	<p>Потсетување за еднодимензионални низи</p> <p>Алгоритми за пребарување</p> <ul style="list-style-type: none"> <li>- Линеарно пребарување и неговата сложеност</li> <li>- Бинарно пребарување и негова сложеност</li> </ul> <p>Сортирачки алгоритми</p> <ul style="list-style-type: none"> <li>- Сортирање со селекција и негова сложеност (анг. selection sort)</li> </ul>	<p>Пребарување</p> <p>Пребарувачки клуч</p> <p>Сортирање</p> <p>Сортирачки клуч</p> <p>Линеарно пребарување</p> <p>Бинарно пребарување</p> <p>Логаритамско време на извршување</p> <p>Сортирање со селекција</p> <p>Кофа сортирање</p> <p>Сортирање со</p>	<ul style="list-style-type: none"> <li>- Дискусија за проблеми од секојдневниот живот кои вклучуваат пребарување низ големо количество податоци (телефонски именик, речник и сл.).</li> <li>- Дискусија за времето за извршување на алгоритмот за пребарување, односно бројот на споредби во зависност од големината на векторот.</li> <li>- Дискусија и презентација на алгоритми за пребарување (пр. Тестирање дали првиот елемент од низа е еднаков на некој друг елемент, тестирање дали постојат два исти елементи во низата и сл.).</li> <li>- Презентација на алгоритмот за бинарно пребарување.</li> <li>- Дискусија за условите кои треба да ги исполнува низата за да може да се користи бинарно пребарување.</li> <li>- Дискусија за сложеноста на алгоритмот за бинарно пребарување.</li> <li>- Изработка на програми кои користат бинарно пребарување.</li> <li>- Дискусија за проблеми од секојдневниот живот кои вклучуваат сортирање на податоци (пр. сортирање по азбучен редослед во дневник и сл.).</li> </ul>





<p>алгоритмите за пребарување прво по широчина и пребарување прво по длабочина.</p>	<p>на матрици</p> <p>Пребарување низ графови</p> <ul style="list-style-type: none"> <li>- Пребарување прво по широчина</li> <li>- Пребарување прво по длабочина</li> </ul> <p>Решавање на проблеми со графови</p>	<p>Сврзаност</p> <p>Пребарување прво по широчина</p> <p>Пребарување прво по длабочина</p>	<p>пребарување прво по широчина.</p> <ul style="list-style-type: none"> <li>- Дискусија за проблеми кои се решаваат со алгоритмот за пребарување прво по широчина (најкратка оддалеченост меѓу две темиња и сврзаност на граф).</li> <li>- Илустрација на решенија на проблеми кои го користат алгоритмот за пребарување прво по широчина (проблеми кои можат да се претстават со матрици како најкраток пат низ лавиринт и сл.).</li> <li>- Дискусија и презентација на алгоритмот за пребарување прво по длабочина.</li> <li>- Илустрација на решенија на проблеми кои го користат алгоритмот за пребарување прво по длабочина, Backtracking (преку проблеми кои можат да се претстават со матрици).</li> </ul>
---	---	---	---

### Програмско подрачје 3: Показувачи (8 часа)

<i>Цели</i>	<i>Содржини</i>	<i>Поими</i>	<i>Активности и методи</i>
<p>Ученикот/ученичката:</p> <ul style="list-style-type: none"> <li>- да го осознае концептот на показувачите и потребата од нивно користење;</li> <li>- да знае да декларира и иницијализира показувачка променлива од различен тип, да „оживее“ показувач со new;</li> <li>- да знае да користи оператори за показувачи;</li> <li>- да се запознае со можноста за користење на референца преку показувач кај функции;</li> </ul>	<ul style="list-style-type: none"> <li>- Декларација и иницијализација на показувачка променлива од одреден тип</li> <li>- Резервирање на меморија со помош на показувач (new)</li> <li>- Оператори за показувачи</li> </ul>	<p>Показувач (показувачка променлива)</p> <p>Директно и индиректно референцирање променлива</p> <p>Нулти показувач</p> <p>Оператори за показувачи</p> <p>Адресен оператор</p> <p>Оператор за индирекција</p>	<ul style="list-style-type: none"> <li>- Дискусија за сместување во меморијата на статичките и динамичките променливи и потребата од постоење показувачи.</li> <li>- Објаснување за директно и индиректно референцирање променлива преку наведување примери и графички репрезентации.</li> <li>- Презентација на едноставни пример програми кои користат показувачи.</li> <li>- Објаснување на оператори за показувачи преку графички репрезентации и едноставни пример програми.</li> <li>- Објаснување за чести грешки при програмирање кога се користи дереференцирање на показувач (дереференцирање променлива која не е показувач, дереференцирање на неиницијализиран показувач, дереференцирање на нулти показувач).</li> <li>- Дискусија, споредба и истакнување разлики</li> </ul>



<ul style="list-style-type: none"> <li>- да научи да креира низа во динамичка меморија со помош на покажувач;</li> <li>- да ја сфати поврзаноста помеѓу покажувачите и низите;</li> <li>- да знае аритметички операции кои може да се изведуваат со покажувачи;</li> <li>- да ја сфати разликата помеѓу стандардната аритметика и аритметиката со покажувачи;</li> <li>- да се оспособи да изработува програми со правилно користење адресна аритметика;</li> <li>- да увиди дека покажувачите кои покажуваат кон низи може да се индексираат исто како и имињата на низите;</li> <li>- да знае дека сите изрази со индексирани низи може да се запишат со покажувач и поместување;</li> <li>- да увиди дека може да се дефинираат низи од покажувачки променливи;</li> </ul>	<ul style="list-style-type: none"> <li>- Креирање низа во динамичка меморија со помош на покажувач</li> <li>- Поврзаност меѓу покажувачи и низи</li> <li>- Адресна аритметика</li> <li>- Пример програма со низа од покажувачи</li> </ul>	<p>(оператор за дереференцирање)</p> <p>Операторот new</p> <p>Инкрементирање, декрементирање на покажувач, додавање и одземање целобројна вредност, додавање и одземање еден покажувач од друг</p>	<p>помеѓу трите начина за предавање аргументи на функција во C++ - предавање по вредност, по референца со аргументи – референци, по референца со аргументи – покажувачи.</p> <ul style="list-style-type: none"> <li>- Дискусија за креирање низа во динамичка меморија со помош на покажувач и изработка на програми кои го вклучуваат тоа.</li> <li>- Дискусија за тесната поврзаност меѓу покажувачите и низите.</li> <li>- Објаснување преку споредба на аритметички операции во стандардната аритметика и аритметиката со покажувачи.</li> <li>- Дискусија и анализа на пример програми кои содржат аритметика со покажувачи: (инкрементирање (++), декрементирање на покажувач (--), додавање (+ или +=) и одземање (- или -=) целобројна вредност на/од покажувач, додавање и одземање еден покажувач на/од друг).</li> <li>- Дискусија околу чести грешки при програмирање кога се користи аритметиката со покажувачи (логички грешки кои се појавуваат при: користење на аритметика со покажувачи на покажувач кој не покажува кон низа, одземање или споредба на два покажувачи кои не се однесуваат на елементи од иста низа, користење на аритметика со покажувачи за да се помести покажувач надвор од границите на некоја низа).</li> <li>- Изработка на едноставни програми кои вклучуваат адресна аритметика.</li> <li>- Дискусија и објаснување на примери преку кои треба да се согледа дека сите изрази со индексирани низи може да се запишат со покажувач и поместување, користејќи го или името на низата како покажувач или користејќи посебен покажувач кој покажува кон низата.</li> <li>- Објаснување и дискусија за нотацијата покажувач/поместување.</li> <li>- Објаснување како покажувачите се користат за</li> </ul>
---	---	--	--



<p>- да се оспособи за практична примена на стекнатите знаења за покажувачи преку изработка на програми.</p>			<p>пристап до елементи на низа (нотација со индексирани низа, нотација покажувач/поместување со името на низата како покажувач, нотација со индексирани покажувач, нотација покажувач/поместување со покажувач).</p> <p>- Изработка на едноставни програми (пр. печатење на пет елементи од целобројна низа со користење на четирите нотации за пристап до елементите на низа).</p> <p>- Изработка на програми со користење покажувачи.</p>
--	--	--	---

#### Програмско подрачје 4: Класи – подлабок поглед (11 часа)

<i>Цели</i>	<i>Содржини</i>	<i>Поими</i>	<i>Активности и методи</i>
<p>Ученикот/ученичката:</p> <ul style="list-style-type: none"> <li>- да се потсети на концептите на објектно ориентираното програмирање;</li> <li>- да умеет да ги распознава елементите на ООП (класи, објекти, функции-членки, податочни членки) и нивната функција;</li> <li>- да се потсети на синтаксата за креирање на класа и на функцијата на функциите за поставување и преземање;</li> <li>- да ја согледа улогата на деструкторот наспроти улогата на конструкторот;</li> <li>- да се запознае со концептот на композиција;</li> <li>- да се оспособи за имплементација на композиција при креирањето на класи;</li> <li>- да се оспособи за разбирање на поопсежни ОО програми преку</li> </ul>	<p>Повторување за основните концепти и поими од објектно – ориентираното програмирање (ООП): класи, објекти, функции-членки, податочни членки, дефинирање на класа со ф-ја членка, функции за поставување и преземање, конструктори</p> <p>- Студија на случај за дадена класа:</p> <ul style="list-style-type: none"> <li>- Делокруг на класа и пристапување до членките од класата</li> <li>- Функции за пристап и услужни</li> </ul>	<p>Деструктор</p> <p>Покажувач this</p> <p>Композиција</p>	<ul style="list-style-type: none"> <li>- Дискусија за концептите на објектно – ориентираното програмирање, елементите на ООП (класи, објекти, функции-членки, податочни членки) и нивната функција.</li> <li>- Изработка на програми за потсетување на синтаксата за креирање на класа, формата и функцијата на функциите за поставување и преземање како и на улогата на конструкторот.</li> <li>- Запознавање со новите концепти (покажувач this, деструктор, композиција) преку разгледување на опсежна студија на случај.</li> <li>- Изработка на програми со креирање на класи и тестирачки функции за тие класи, кои ќе ги вклучуваат новоизучените концепти.</li> </ul>



<p>разгледување на студија на случај;</p> <ul style="list-style-type: none"> <li>- да се оспособи за примена на изучените концепти во конструкција на класи и изработка на програми .</li> </ul>	<p>функции</p> <ul style="list-style-type: none"> <li>- Конструктори и деструктори</li> <li>- Композиција</li> <li>- Статички членки</li> </ul> <p>на класата</p> <ul style="list-style-type: none"> <li>- Класата String.</li> </ul> <p>Работа со објекти од класата стринг</p>		
--	--	--	--

**Програмско подрачје 5: Датотеки. Текстуални датотеки (7 часа)**

Цели	Содржини	Поими	Активности и методи
<p>Ученикот/ученичката:</p> <ul style="list-style-type: none"> <li>- да знае да дефинира и опишува датотека;</li> <li>- да знае да направи разлика помеѓу датотеки со случаен пристап и текстуални датотеки;</li> <li>- да знае да дефинира и опишува стандардни процедури и функции за работа со текстуални датотеки;</li> <li>- да знае да креира и практично да реализира едноставни програми користејќи текстуални датотеки;</li> <li>- да се оспособи да креира и практично да реализира програми за едноставни операции со текстуални датотеки како: креирање, приказ на содржината од датотеката на екран, префрлање на податоци од една во друга текстуална датотека.</li> </ul>	<p>Датотеки и видови датотеки: датотеки со случаен пристап, датотеки со секвенцијален пристап (текстуални датотеки) Поврзување на поимот за датотека со физичка датотека на хард дискот</p> <p>Променливи за работа со текстуални датотеки</p> <p>Функции за работа со текстуални датотеки</p> <p>Креирање на текстуална датотека преку програма и директно преку некој текстуален едитор</p>	<p>Датотеки</p> <p>Случаен пристап</p> <p>Секвенцијален пристап</p> <p>Текстуални датотеки</p> <p>Променливи за работа со текстуални датотеки</p> <p>Функции за работа со текстуални датотеки</p>	<ul style="list-style-type: none"> <li>- Презентирање на примери од секојдневието каде се применуваат датотеките.</li> <li>- Презентација на пример програми кои користат Датотеки.</li> <li>- Дискусија за тоа каде се сместени датотеките во компјутер и како може текстуалните датотеки да се креираат со текстуален едитор.</li> <li>- Дискусија за синтаксата на функциите (наредбите) коишто се користат во датотеките.</li> <li>- Изработка на програми во кои се вклучени за едноставни операции со текстуални датотеки како: креирање, приказ на содржината од датотеката на екран, префрлање на податоци од една во друга текстуална датотека.</li> <li>- Изработка на програми (може и претходно изработените) во кои влезот ќе се чита од претходно постоечка текстуална датотека, а излезот ќе се запишува во новокреирана</li> </ul>



	Запишување на податоци во текстуална датотека  Читање на податоци од текстуална датотека		текстуална датотека.
--	--	--	----------------------

**Програмско подрачје 6: Податоци од тип структура (struct) (5 часа)**

Цели	Содржини	Поими	Активности и методи
Ученикот/ученичката:  - да знае да дефинира, декларира и иницијализира податочен тип структура;  - да ја увиди потребата од користење на податочниот тип структура;  - да знае да декларира покажувач кон податок од тип структура;       - да се оспособи за примена на податоци од тип структура (само со податочни членки) во програми.	Дефинирање на податоци од тип структура  Декларирање и иницијализација на променливи од даден тип структура  Покажувачи кон податоци од тип структура  Функции-членки на структурата  Изработка на програми	Структура како тип на податок, struct  Членови на структура  Анонимна структура   Променливи од соодветен тип структура    Оператор за дофат	- Дискусија за структурирани типови податоци (низи, стрингови) и поврзување на претходните знаења со новиот податочен тип структура. - Дискусија и давање примери за примената на податочниот тип структура. - Дефинирање на новиот податочен тип – struct. - Дискусија и објаснување за потребата за уникатност на имињата на членовите од иста структура, но не и од различни структури. - Дефинирање на променливи од даден тип структура. - Дискусија и објаснување за пристап до членовите на структурата преку презентирање примери. - Презентирање на примери за структура и воочување на честа грешка при програмирање (синтаксичка) – заборавање на точка запирачка на крајот од дефиницијата на структура. - Објаснување пристап до членови на структура преку покажувач. - Изработка на едноставни програми со примена на стекнатите знаења за податочен тип структура.

**Програмско подрачје 7: Податочни структури (12 часа)**

Цели	Содржини	Поими	Активности и методи
Ученикот/ученичката: - да умеа да објасни и да даде пример за прости и сложени	Динамички структури на податоци	Прости и сложени структури од	- Дискусија, објаснување, поврзување на претходно стекнатите знаења за типови податоци со новите поими.



<p>структури од податоци;  - да го разбере поимот динамичка структура на податоци;  - да ја знае разликата помеѓу статички и динамички структури на податоци;  - да умеа да дефинира самореференцирачки тип структура;</p> <p>- да го знае поимот линеарна поврзана листа;  - да умеа да го објасни поимот еднострано поврзана линеарна листа;  - да го препознава поимот јазол (елемент на едностранна листа) и да објаснува неговата структура;  - да се оспособи за креирање подалгоритми за основните операции со еднострано поврзана линеарна листа;</p> <p>- да умеа да го објасни поимот куп  - да се оспособи за креирање подалгоритми за основните операции со куп;</p> <p>- да умеа да го објасни поимот ред и да умеа да направи разлика од куп;  - да умеа да го објасни поимот дрво како нелинеарна листа;</p> <p>- да се запознае со пример програми во кои се применуваат погоренаведените типови на податочни структури;</p>	<p>Самореференцирачки тип структура</p> <p>Динамичка алокација на меморија и податочни структури</p> <p>Линеарни поврзани листи  Еднострани поврзани линеарни листи (креирање и манипулација со елементите)</p> <p>Куп (анг. stack) и основни операции кај куп (креирање и манипулација со елементите)</p> <p>Запознавање со ред (анг. queue) и основни операции кај ред наспроти куп</p> <p>Нелинеарни наспроти линеарни податочни структури, пример: дрво</p>	<p>податоци</p> <p>Статички и динамички структури на податоци апстрактни структури</p> <p>Еднострано поврзани листи  Јазол (елемент на листа)  Покажувач кон јазол</p> <p>Куп  Врв на куп  Избивање елемент од куп  Набивање елемент во куп</p> <p>Ред  Опашка и глава на редот  Покажувач на почетокот/крајот на редот  Кружен ред</p> <p>Дрво</p>	<p>- Објаснување и дискусија за статички и динамички податочни структури.</p> <p>- Објаснување преку примери за самореференцирачки тип структура  - Дискусија за динамичка алокација на меморија и податочни структури.</p> <p>- Објаснување за јазол и за структура на еднострано поврзана линеарна листа преку графички прикази.  - Креирање и анализа на подалгоритми за операции со еднострано поврзани линеарни листи: креирање празна еднострано поврзана линеарна листа, додавање јазол на почетокот/крајот на еднострано поврзана линеарна листа, додавање јазол помеѓу два јазли во еднострано поврзана линеарна листа, бришење јазол од еднострано поврзана линеарна листа.</p> <p>- Објаснување и дискусија за куп преку графички приказ на куп со неколку елементи.  - Креирање и анализа на подалгоритми за основните операции (иницијализација на празен куп, проверка дали купот е полн, ставање податок во куп, проверка дали купот е празен и земање податок од куп) за имплементација на куп.</p> <p>- Објаснување и дискусија за ред преку графички приказ на ред со неколку елементи.  - Презентација на подалгоритми за основните операции (иницијализација на празен ред, проверка дали редот е полн, ставање податок на крајот од редот, проверка дали редот е празен и земање податок од почетокот на редот) за имплементација на ред.</p> <p>- Објаснување и дискусија преку шематски приказ</p>
--	---	---	---



- да се оспособи за креирање на програми со примена на еднострано поврзана листа и куп како типови на податочни структури.	Изработка на креирање на програми со примена на погоренаведените типови на податочни структури		за елемент (јазол) на дрво - Изработка на креирање на програми со примена на еднострано поврзана листа и куп како типови на податочни структури.
--	--	--	---

### Програмско подрачје 8: Стандардна библиотека на шаблони (12 часа)

Цели	Содржини	Поими	Активности и методи
<p>Ученикот/ученичката:</p> <ul style="list-style-type: none"> <li>- да се запознае со множеството од структури и алгоритми (STL - Standard Template Library);</li> <li>- да се запознае со операциите кои можат да се вршат врз различни типови податоци;</li> <li>- да ги разбере составните делови на стандардната библиотека на шаблони: контејнерите, итераторите и алгоритмите;</li> <li>- да ги утврди претходните знаења за класен шаблон &lt;vector&gt; од стандардната библиотека на C++;</li> <li>- да се оспособи да решава проблеми со употреба на класните шаблони и нивните методи;</li> <li>- да се запознае со пример за асоцијативен контејнер - мапа;</li> <li>- да умее да решава програмски проблеми со употреба на мапа;</li> <li>- да се запознае со пример за контејнер адаптер - куп;</li> <li>- да умее да решава програмски проблеми со употреба на</li> </ul>	<p>Стандардната библиотека на шаблони: контејнери, итератори и алгоритми</p> <p>Контејнери и нивна поделба (секвенцијални, асоцијативни, адаптери)</p> <p>Креирање празен вектор, додавање елементи и печатење на содржина</p> <p>Примена на класен шаблон &lt;vector&gt; од стандардната библиотека на C++ како замена за еднодимензионална низа</p> <p>Употреба на итератори</p>	<p>Стандардната библиотека на шаблони</p> <p>Контејнери итератори алгоритми</p> <p>Секвенцијални контејнери</p> <p>Асоцијативни контејнери</p> <p>Контејнери адаптери</p> <p>vector map stack push_back front back size reverse sort resize</p>	<ul style="list-style-type: none"> <li>- Дискусија за стандардната библиотека на шаблони на C++.</li> <li>- Објаснување на множеството структури од стандардната библиотека на шаблони на C++.</li> <li>- Објаснување на множеството компоненти кои имплементираат голем број на познати податочни структури и алгоритми за обработка на тие структури.</li> <li>- Практична изработка на програми кои сместуваат податоци од ист и различен тип.</li> <li>- Пристапување до елементите во даден вектор и употреба на push_back, front, back, size, reverse, resize и clear.</li> <li>- Изработка на програми кои вклучуваат методи begin, end, insert и erase.</li> <li>- Разгледување на примери и задачи со итератори.</li> <li>- Изработка на програми со примена на мапа.</li> <li>- Изработка на програми со примена на куп.</li> </ul>



контејнер адаптер - куп; - да се оспособи правилно да користи елементи од множеството од структури и алгоритми (STL - Standard Template Library).	Примена на асоцијативен контејнер мапа (map)  Примена на контејнер адаптер: куп (stack)	clear  Итератор push pop	
--	---	--------------------------------------	--

### Програмско подрачје 9: Настан-базирано програмирање (12 часа)

Цели	Содржини	Поими	Активности и методи
Ученикот/ученичката:  - да се стекне со теоретски познавања од настан-базирано (визуелно) програмирање; - со практично искуство да се запознае со алатка за визуелен брз развој на апликации; - да стекне основни практични познавања во визуелно програмирање на апликации со графички кориснички интерфејс; - да се оспособи за изработка на едноставни програми со графички кориснички интерфејс; - да умее да креира форми и компоненти и да поставува и менува својства на компоненти; - да поставува соодветни имиња на избраните компоненти и да манипулира со нивните можности во визуелна програма; - да знае да ракува со настани при извршување на одредени надворешни настани; - да умее да примени претходно изучувани програмски техники	Компоненти на интегрирана развојна околина (IDE) за настан-базирано (визуелно) програмирање, едитор, компајлер, поврзувач, библиотеки  Запознавање со основните елементи на околина (која ќе се користи) за визуелен развој  Визуелно програмирање – основни визуелни контроли (форми и работа со повеќе форми, копчиња, менија) и настани.  Креирање нов проект, поставување и	Настан-базирано (визуелно) програмирање  Форми контроли својства на форми и контроли  Лента со алатки - Toolbox  Копче (анг. Button) копче со избор етикета (анг. label) поле за текст радио копче поле со листа панел  ракувач со настани  trigger функции  .NET рамка	- Дискусија за практична примена на настан-базирани (визуелни) програми.  - Презентација на претходно креирани програмски решенија со визуелни компоненти.  - Креирање на нов проект и поставување на основни визуелни компоненти. - Уредување на избраните компоненти.   - Поставување настани (англ. events) на поставените компоненти. - Ракување со настани извршени при одредени надворешни настани.  - Креирање едноставна програма со поле за текст



<p>при креирање на визуелни програмски решенија.</p>	<p>уредување на визуелни компоненти</p> <p>Лента со алатки - Toolbox и нејзини можности</p> <p>Поставување настани на визуелни компоненти</p> <p>Креирање визуелни програми (GUI апликации) кои извршуваат одредена активност</p>	<p>GUI апликација</p> <p>CLR Forms Application</p>	<p>и етикета и испишување на соодветна порака во етикета.</p> <p>- Изработка на мини проекти со вклучување на доделувања, аритметички пресметки, читање на вредности од тастатура и прикази на резултати во етикета (пр: внес на два броја со соодветна операција и резултат во етикета, конвертор на валути, калкулатор.</p> <p>- Разгледување на готови програми.</p> <p>- Креирање на визуелни програми кои решаваат ситуации од секојдневниот живот.</p>
--	---	--	--



#### 4. ДИДАКТИЧКИ ПРЕПОРАКИ

- Насоки за меѓупредметно поврзување (интегрирано планирање, односно целно и содржинско поврзување меѓу сродни предмети и подрачја):

- мајчин јазик (синтакса, семантика и граматика);
- математика (цели броеви, реални броеви, координатен систем, геометриски тела, формули, функции и работа со различни типови на податоци);
- информатика (оперативни системи, програма за цртање, програма за уредување текст и програма за табеларни пресметки);
- ликовно образование (креативно и визуелно осмислување кај некои задачи);
- сите останати предмети чии содржини претставуваат содржини на изработување на проектни задачи.

Препораки за користење на годишниот фонд на часовите

За секое предвидено програмско подрачје предложен е одреден годишен фонд на часови, односно:

- Функциски шаблони, рекурзија и рекурзивни функции - 10 часа;
- Алгоритми кај повеќе-димензионални низи и графови – 22 часа;
- Показувачи – 8 часа;
- Класи - 11 часа;

- Датотеки. Текстуални датотеки - 7 часа;
- Податоци од тип структура(struct) – 5 часа;
- Податочни структури – 12 часа;
- Стандардна библиотека на шаблони - 12 часа;
- Настан-базирано програмирање – 12 часа.

Од предвидениот вкупен фонд на часови по наставните програмски подрачја дозволени се мали отстапувања согласно знаењата на учениците, претходната подготовка на учениците за работа со компјутер, дополнителната опрема со која располага училиштето и слично. Во овој контекст, некои содржини ќе се повторуваат повеќе пати за сметка на други кои ќе се работат со помал фонд на часови.

Наставни средства:

- учебник за програмски јазици избран на ниво на училиштето, одобрен од министерот;
- литература: се препорачува користење на учебник од проектот на Владата „Превод на 500 стручни научни книги и учебници“, Пол Дејтел, Харви Дејтел: **C++: КАКО СЕ ПРОГРАМИРА**, глава 6 - Функции и вовед во рекурзија, глава 7- Низи и вектори, глава 8 – Показувачи, глава 9 – Класи: подлабок поглед, прв дел, глава 10 – Класи: подлабок поглед, втор дел, глава 17 – Процесирање датотеки, глава 18 – Класата string и процесирање на поток од објект од string, глава 19 – Пребарување и сортирање, глава 20 – Податочни структури, глава 21 – Битови, знаци, C – стрингови и податоци од тип struct и глава 22 – Стандардна шаблонска библиотека (STL), Арс ламина, Скопје 2010
- Интернет, образовни софтвери;
- Интегрирана околина за програмирање и демо програми;
- списанија;
- аудио-визуелни средства.

## 5. ОЦЕНУВАЊЕ НА ПОСТИГАЊАТА НА УЧЕНИЦИТЕ

Во текот на наставата по информатика се препорачува формативно следење кое вклучува изработка и водење портфолио на учениците што опфаќа:

- собирање на показатели (ученички изработки на компјутер) за секој ученик посебно;
- тековни (формативни) однапред подготвени евалвациони листи за секој ученик посебно кои се изработуваат по конкретната негова активност.

На крајот на секое тримесечје, врз основа на сознанијата од формативното оценување се реализира микросумативното оценување. Согласно природата на програмата по предметот програмски јазици оценувањето може да се реализира усно, практично, со презентација и слично.

Ученикот се оценува со бројчана оценка.



## 6. ПРОСТОРНИ УСЛОВИ ЗА РЕАЛИЗИРАЊЕ НА НАСТАВНАТА ПРОГРАМА

Програмата во однос на просторните услови се темели на Нормативот за простор, опрема и наставни средства за средно образование донесен од страна на министерот за образование и наука.

## 7. НОРМАТИВ ЗА НАСТАВЕН КАДАР

- Завршени студии по:
- информатика VII/1, наставна насока или применета насока со педагошко-психолошка и методска подготовка;
- математика-информатика VII /1, наставна насока;
- компјутерска техника, информатика и автоматика, VII /1 со педагошко-психолошка и методска подготовка;
- завршени дипломски четиригодишни студии од областа на информатиката, информациските или информациско - комуникациските технологии на акредитираните универзитети во РМ или во странство со домашна акредитација и со педагошко-психолошка и методска подготовка.

## 8. ОЧЕКУВАНИ РЕЗУЛТАТИ

Ученикот/ученичката:

- знае поими, правила, постапки и процедури за изработка на алгоритми и програми со користење на програмскиот јазик C++;
- користи правилно интегрирани околинати за програмирање;
- умее да изготви продукти со програмскиот јазик C++;
- знае да истражува и да ја оценува точноста, релевантноста, соодветноста и можностите на електронските информациски извори;
- умее да применува правила, постапки и процедури за решавање на одредени проблеми;
- ја планира и ја организира својата работа на конкретна задача со компјутер;
- комбинира различни техники за програм при реализација на одредена задача;
- стекнатото знаење за работа на компјутер може да го користи за креативно изготвување на продукти;
- истражува и пронаоѓа начини за изработка на текстови, цртежи, презентации, тестови, готови продукти и слично со едукативна, контролна, креативна и/или забавна содржина со користење на корисничките програми;
- комбинира елементи во нова целина;
- изготвува целосен проект и готов продукт.

## 9. ПОЧЕТОК НА ПРИМЕНА НА НАСТАВНАТА ПРОГРАМА

Датум на започнување: 1.09.2014 година



## 10. Потпис и датум на утврдување на наставната програма

Наставната програма по *програмски јазици*, избран предмет во природно-математичко подрачје А, четврта година на гимназиското образование., на предлог на Бирото за развој на образованието, ја утврди

  
РЕПУБЛИКА МАКЕДОНИЈА  
МИНИСТЕРСТВО ЗА ОБРАЗОВАНИЕ И НАУКА  
Бр. 11-145991  
10-09-2016 год.  
СКОПЈЕ

Министер  
Abdilaqim Ademi



Изготвил: работна група формирана со решение бр. 02-579/1 од 27.06.2014 година, координатор Гордана Јанакиевска, раководител на одделение  
Контролирал: Трајче Георгиевски, раководител на одделение  
Одобрил: м-р Митко Чешларов, раководител на сектор

