

Врз основа на член 55 став 1 од Законот за организација и работа на органите на државната управа („Службен весник на Република Северна Македонија“ бр. 58/00, 44/02, 82/08,167/10,51/11,96/2019 и 110/2019) и член 22 став 1 од Законот за средно образование („Службен весник на Република Северна 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, 18/11,42/11, 51/11, 6/12, 100/12, 24/13, 41/14, 116/14, 135/14, 10/15, 98/15, 145/15, 30/16, 127/16, 67/17 64/18 и 229/20) и член 3 од Законот за математичко-информатичка гимназија („Службен весник на Република Северна Македонија“ бр. 64/18), министерот за образование и наука ја донесе Наставната програма по предметот *напредно програмирање* за III (трета) година математичко-информатичка гимназија.

МИНИСТЕРСТВО ЗА ОБРАЗОВАНИЕ И НАУКА
БИРО ЗА РАЗВОЈ НА ОБРАЗОВАНИЕТО



Наставна програма
НАПРЕДНО ПРОГРАМИРАЊЕ
модуларно дизајнирана

за III година

Математичко-информатичка гимназија

Скопје, 2022 година

Назив на наставната програма	Напредно програмирање
Тип на наставна програма	Изборна
Кредитна вредност на наставната програма	5 (пет) ЕЦВЕТ ¹ кредити
Ниво на квалификација	IV(четврто) ниво
Година на изучување	III (трета)
Број на часови неделно/годишно за реализација на наставната програма	3/108
Цели на наставна програма	<p>Ученикот/ученичката:</p> <ul style="list-style-type: none"> - да ги продлабочи знаењата по информатика во областа на програмирањето и решавањето на проблеми и да ги применува стекнатите знаења во секојдневни ситуации, во соодветни предизвици како и во други наставни предмети. - да постигне самодоверба во примена на стекнатите програмерски и алгоритамски вештини за наоѓање, користење и презентирање на математичките аргументи. - да ја цени убавината, моќта, корисноста и интернационалната димензија на информатиката и програмирањето и да извлекува задоволство од постигнатите резултати. - да развива логичко, критичко и креативно алгоритамско размислување. - да се подготви за успешни учество на врвни натпревари од областа на програмирањето.

¹Закон за Националната рамка на квалификации.

<p>Модуларни единици на наставна програма</p>	<ul style="list-style-type: none"> • ЕЛЕМЕНТИ НА КОМПЕТИТИВНО ПРОГРАМИРАЊЕ (15 часа) • ФОРМАЛНИ АЛГОРИТМИ (18 ЧАСА) • ФОРМАЛНИ ЈАЗИЦИ И АВТОМАТИ (21 ЧАС) • ТЕОРИЈА НА ПРЕСМЕТКОВНА КОМПЛЕКСНОСТ (6 ЧАСА) • NP-ТЕШКИ/КОМПЛЕТНИ ПРОБЛЕМИ (12 ЧАСА) • ДЕКОМПОЗИЦИЈА НА ПРОБЛЕМИ (12 ЧАСА) • СПЕЦИФИЧНИ ПРОБЛЕМСКИ ЗАДАЧИ (24 ЧАСА)
<p>Материјално-технички и просторни услови</p>	<p>За постигнување на целите на наставата по програмирање неопходно е стручно осмислена и планирана примена на различни наставни средства и задолжително компјутер за секој ученик, со соодветно инсталирани програмски пакети и прилагодени привилегии за корисникот, поврзани на интернет. Наставникот треба да поседува преносен компјутер и опрема за проектирање.</p>
<p>Норматив на наставен кадар</p>	<p>Наставната програма може да ја реализира стручно лице кое има познавање од англиски јазик и вештини за користење компјутер односно:</p> <ul style="list-style-type: none"> - наставник со завршени студии по информатика/ наставна или друга насока VII/1 или VI A според НРК и 240 ЕКТС; <p>Стручното лице треба да исполнува најмалку еден од следните услови:</p> <ul style="list-style-type: none"> - да бил ментор на ученик кој бил награден на престижен меѓународен натпревар од соодветната област; - научен степен доктор на информатички науки; - да е запишан на докторски студии на соодветната област. - да има стекнато научен степен доктор на науки на соодветната област.

Модуларна единица 1: ЕЛЕМЕНТИ НА КОМПЕТИТИВНО ПРОГРАМИРАЊЕ (15 часа)

Ред. број	Резултати од учење	Содржини и поими	Активности и методи	Критериуми на оценување*
1	<p>Ученикот/ученичката ќе биде способен/а :</p> <ul style="list-style-type: none"> - да ги наброи врвните натпревари во програмирање и да ги опише нивните формати; - да решава проблемски задачи во услови на точно дефиниран формат на влез и излез, временско ограничување и ограничена временска и просторна сложеност на решението; - да имплементира програмски решенија на ат-хок проблеми. 	<ul style="list-style-type: none"> • Вовед во компетитивно програмирање • Преглед на врвни натпревари во програмирање и проучување на нивните формати • Методи за подобра компетитивност - Побрзо типкање на код - Брза идентификација на типот на проблем - Анализа на алгоритмот - Усовршување на познавањето на програмскиот јазик - Усовршување на вештината на тестирање на код - Редовен тренинг - Тимска работа 	<p>Активности:</p> <ul style="list-style-type: none"> • Наставникот прави презентација на целите на курсот и дава вовед во компетитивно програмирање • Учениците истражуваат информации за најпрестижните натпревари во програмирање во светот и прават нивна компарација • Наставникот ги презентира еден по еден методите за подобра компетитивност, а учениците решаваат соодветни множества задачи • Наставникот ја објаснува анатомијата на една натпреварувачка задача и форматите за влез/излез на податоци 	<p>Ученикот/ученичката може да:</p> <p>1.1: објасни и спореди методи за подобра компетитивност;</p> <p>1.2: применува компетитивни техники во програмирањето;</p> <p>1.3: запазува временско ограничување и ограничена временска и просторна сложеност во кодот на решението;</p> <p>1.4: изработува програми во услови на точно дефиниран формат на влез и излез;</p> <p>1.5: учествува во тимска работа за решавање проблеми;</p> <p>1.6: креира програми за решавање на ат-хок проблеми.</p>

*Внесени се стандарди/индикатори за постигнување на резултатите од учењето врз основа на кои се определуваат критериумите за оценување.

		<ul style="list-style-type: none"> • Анатомија на натпреварувачка задача - Типични влезно излезни рутини • Решавање на ат-хок проблеми <p>Поими: Компетитивно програмирање, Ат-хок проблем</p>	<ul style="list-style-type: none"> • Учениците имплементираат решенија на избрано множество ат-хок задачи <p>Методи: дискусија, дијалог, презентација, учење преку откривање-истражување, бура на идеи, практична работа на компјутер - решавање на проблемски задачи</p>	
Модуларна единица 2: ФОРМАЛНИ АЛГОРИТМИ (18 ЧАСА)				
Ред. број	Резултати од учење	Содржини и поими	Активности и методи	Критериуми на оценување*
1	<p>Ученикот/ученичката ќе биде способен/а :</p> <ul style="list-style-type: none"> - да ги дефинира основните термини кај формални алгоритми - да користи Тјурингови машини за решавање соодветни проблеми - да користи Нормални алгоритми за решавање соодветни проблеми 	<ul style="list-style-type: none"> • Запознавање со основните термини - Азбуки, букви, зборови - Функции, терми, алгебри • Запознавање со тјурингови машини • Решавање на проблеми со тјурингови машини • Запознавање со нормални алгоритми • Решавање задачи со нормални алгоритми 	<p>Активности:</p> <ul style="list-style-type: none"> • Наставникот прави презентација на секоја содржина соодветно • Наставникот презентира начини на решавање на задачи од соодветната содржина • Учениците решаваат задачи од избрано множество за секоја содржина соодветно • Учениците истражуваат за примери проблемски задачи 	<p>Ученикот/ученичката може да:</p> <p>1.1: ги објасни основните термини кај формални алгоритми;</p> <p>1.2 ги наброи елементите на Тјурингова машина</p> <p>1.2 објасни за Нормални алгоритми и да наведе проблеми кои може да се решат со нивна примена</p> <p>1.3: креира решенија на проблеми со Тјурингови машини;</p> <p>1.4: креира Нормални алгоритми за решавање на соодветни проблеми;</p>

*Внесени се стандарди/индикатори за постигнување на резултатите од учењето врз основа на кои се определуваат критериумите за оценување.

	<p>- да користи (примитивно) рекурзивни функции за решавање соодветни проблеми да решава проблемски задачи кои во себе ги кријат концептите на изучените формални елементи</p>	<ul style="list-style-type: none"> • Оператори за супституција и примитивна рекурзија • Примитивно рекурзивна функција • Решавање задачи со примитивни рекурзивни функции • Оператор за минимизација • Рекурзивна и општо рекурзивна функција • Решавање на задачи со рекурзивни функции • Решавање на проблемски задачи кои во себе ги кријат концептите на изучените формални елементи <p>Поими: Азбуки, букви, зборови, функции, терми, алгебри, тјурингови машини, нормални алгоритми, оператори за супституција, примитивна рекурзија, примитивно рекурзивна функција, оператор за</p>	<p>кои во себе ги кријат концептите на изучените формални елементи и имплементираат нивни програмски решенија</p> <p>Методи: дискусија, дијалог, презентација, учење преку откривање-истражување, бура на идеи, решавање задачи, практична работа на компјутер - решавање на проблемски задачи</p>	<p>1.5: користи рекурзивни функции при решавање на соодветни проблеми; 1.6: развива програмски решенија на проблемски задачи кои содржат формални елементи.</p>
--	--	--	---	--

		минимизација, општо рекурзивна функција		
Модуларна единица 3: ФОРМАЛНИ ЈАЗИЦИ И АВТОМАТИ (21 ЧАС)				
Ред. број	Резултати од учење	Содржини и поими	Активности и методи	Критериуми на оценување*
1	<p>Ученикот/ученичката ќе биде способен/а :</p> <ul style="list-style-type: none"> - да користи формални модели на пресметливост како основа за теорија и пракса на програмските јазици - креира и употребува регуларни и контекстно слободни јазици, конечни автомати и push down автомати - да користи граматика за дефинирање на јазик - да решава проблемски задачи кои во себе ги користат концептите на формалните јазици и автомати 	<ul style="list-style-type: none"> • Регуларни јазици и регуларни изрази • Решавање задачи со регуларни јазици и регуларни изрази • Детерминистички конечни автомати (ДКА) • Недетерминистички конечни автомати (НКА) • Еквиваленција на НКА со ДКА • Решавање задачи со конечни автомати • Својства на јазици препознаени од конечен автомат • Конечни автомати и регуларни јазици • Решавање на проблемски задачи кои 	<p>Активности:</p> <ul style="list-style-type: none"> • Наставникот прави презентација на секоја содржина соодветно • Наставникот презентира начини на решавање на задачи од соодветната содржина • Учениците решаваат задачи од избрано множество за секоја содржина соодветно • Учениците истражуваат за примери проблемски задачи кои во себе ги кријат концептите на изучените формални елементи и имплементираат нивни програмски решенија <p>Методи: дискусија, дијалог, презентација, учење преку</p>	<p>Ученикот/ученичката може да:</p> <p>1.1: објаснува и дефинира хиерархија на формални јазици;</p> <p>1.2: користи регуларни и контекстно слободни јазици и конечни автомати;</p> <p>1.3: создава решенија на задачи со конечни автомати;</p> <p>1.4: објасни за набивачки автомати и нивна разлика од конечни автомати</p> <p>1.5: креира решенија на задачи со набивачки автомати;</p> <p>1.6: објаснува за формална граматика, и основни елементи на нејзино дефинирање</p> <p>1.7: креира решенија на задачи со граматика;</p> <p>1.8: изработува решенија на проблемски задачи кои во себе ги кријат концептите на контекстно слободни јазици и набивачки автомати.</p>

*Внесени се стандарди/индикатори за постигнување на резултатите од учењето врз основа на кои се определуваат критериумите за оценување.

		<p>во себе ги кријат концептите на регуларни јазици и конечни автомати</p> <ul style="list-style-type: none"> • Типови граматика и јазици генерирани од нив • Регуларни и контекстно слободни јазици • Решавање задачи со контекстно слободни граматика • Набивачки (push-down) автомати • Набивачки (push-down) автомати и контекстно слободни граматика • Својства на контекстно слободните јазици • Решавање задачи со набивачки автомати • Решавање на проблемски задачи кои во себе ги кријат концептите на контекстно слободни јазици и набивачки автомати <p>Поими:</p>	<p>откривање-истражување, бура на идеи, решавање задачи, практична работа на компјутер - решавање на проблемски задачи</p>	
--	--	---	--	--

		<p>Регуларни јазици, регуларни изрази, детерминистички конечни автомати (ДКА), Недетерминистички конечни автомати (НКА), граматики, контекстно слободни јазици, набивачки (push-down) автомати</p>		
--	--	--	--	--

Модуларна единица 4: ТЕОРИЈА НА ПРЕСМЕТКОВНА КОМПЛЕКСНОСТ (6 ЧАСА)				
Ред. број	Резултати од учење	Содржини и поими	Активности и методи	Критериуми на оценување*
1	<p>Ученикот/ученичката ќе биде способен/а :</p> <ul style="list-style-type: none"> - да ги објасни основните идеи и поими во теоријата на пресметковна комплексност 	<ul style="list-style-type: none"> • Вовед во теорија на пресметковна сложеност • Пресметковни проблеми - Проблемски инстанци и нивно репрезентирање - Одлучливи проблеми како формални јазици - Функциски проблеми 	<p>Активности:</p> <ul style="list-style-type: none"> • Наставникот прави презентација на секоја содржина соодветно • Учениците истражуваат за споменатите поими • Дискусија за секој поим и концепт 	<p>Ученикот/ученичката може да:</p> <p>1.1: објаснува и опишува домен на теорија на пресметковна комплексност; 1.2: наброи категории на пресметковни проблеми; 1.3: ги објасни класите на проблеми, P, NP, NP-hard, NP-complete;</p>

*Внесени се стандарди/индикатори за постигнување на резултатите од учењето врз основа на кои се определуваат критериумите за оценување.

		<ul style="list-style-type: none"> • Машински модели и мерки за сложеност • Класи на проблеми, P, NP, NP-hard, NP-complete • P наспроти NP • Неисследливост (intractability) • Неодлучлив проблем <p>- Halting problem</p> <p>Поими: Пресметковна сложеност, Одлучливи проблеми, Функцииски проблеми, P-проблем, NP проблем, NP-hard, NP-complete, Неисследливост (intractability), Неодлучлив проблем, Halting problem</p>	<p>Методи: дискусија, дијалог, презентација, учење преку откривање-истражување, бура на идеи</p>	<p>1.4: објасни за неисследливост и неодлучливост.</p>
--	--	---	---	--

Модуларна единица 5: NP-ТЕШКИ/NP-КОМПЛЕТНИ ПРОБЛЕМИ (12 ЧАСА)

Ред. број	Резултати од учење	Содржини и поими	Активности и методи	Критериуми на оценување*
1	<p>Ученикот/ученичката ќе биде способен/а:</p> <ul style="list-style-type: none"> - Да објасни за пристапите за решавање на NP-тешки проблеми со мали ограничувања во ограничено време за работа; - да наброи различни NP-тешки проблеми, да ги објасни нив, и нивната временска сложеност; - да препознае дека познат NP-тежок проблем може да биде трансформиран до одреден проблем од интерес. 	<ul style="list-style-type: none"> • Пристапи за решавање на NP-тешки проблеми со мали ограничувања во ограничено време за работа • Класични примери на NP-тешки проблеми - Проблем на трговски патник (стандарден наспроти битонски) - Хамилтонов пат - Најдолг прост пат во граф - Максимално независно множество во граф и Минимално покривање на темиња - Покривка со минимален број на патишта (min-path-cover) 	<p>Активности:</p> <ul style="list-style-type: none"> • Наставникот преку презентација и дискусија, со наведување на примери ги запознава учениците со пристапите за решавање на NP-тешки проблеми со мали ограничувања во ограничено време за работа • Наставникот наведува и објаснува решенија на пример NP-тешки проблеми и ја дискутира нивната временска и просторна сложеност • Учениците имплементираат решенија за дадените пример проблеми • Учениците истражуваат за пример NP-тешки проблем 	<p>Ученикот/ученичката може да:</p> <p>1.1: ги опише и дефинира пристапите за решавање на NP-тешки проблеми со мали ограничувања во ограничено време за работа;</p> <p>1.3: наброи класични примери на NP-тешки проблеми;</p> <p>1.4: утврдува временска и просторна сложеност кај решенија; на пример NP-тешки проблеми</p> <p>1.3: имплементира решенија на различни примери на NP-тешки проблем.</p>

*Внесени се стандарди/индикатори за постигнување на резултатите од учењето врз основа на кои се определуваат критериумите за оценување.

		<ul style="list-style-type: none"> - Проблем на стејнерово дрво во граф - Боење на графови - ... • Решавање на избрани проблемски задачи кои се (под)верзии на изучените NP-тешки проблеми <p>Поими: Битонски проблем на трговски патник, Најдолг прост пат во граф, Максимално независно множество во граф, Минимално прокривање на темиња, Покривка со минимален број на патишта (min-path-cover), Проблем на стејнерово дрво во граф</p>	<p>Методи: дискусија, дијалог, учење преку откривање-истражување, практична работа на компјутер - решавање на програми</p>	
--	--	--	---	--

Модуларна единица 6: ДЕКОМПОЗИЦИЈА НА ПРОБЛЕМИ (12 ЧАСА)

Ред. број	Резултати од учење	Содржини и поими	Активности и методи	Критериуми на оценување*
1	<p>Ученикот/ученичката ќе биде способен/а:</p> <ul style="list-style-type: none"> - Да креира програмски решенија за решавање проблеми кои се комбинација од 2 или повеќе алгоритми и/или податочни структури 	<ul style="list-style-type: none"> • Пристапи за решавање проблеми кои се комбинација од 2 или повеќе алгоритми и/или податочни структури • Категории на проблеми кои се комбинација - Двокомпонентен проблем: Бинарно барање на одговор и друга компонента (Алчен алгоритам, геометриска формула,...) - Двокомпонентен проблем: Исполнување на статичка еднодимензионална низа и многу прашанки за опсег на сума/мин/макс - Двокомпонентен проблем: Претпроцесирање на 	<p>Активности:</p> <ul style="list-style-type: none"> • Наставникот преку презентација и дискусија, со наведување на примери ги запознава учениците со пристапите за решавање на проблеми кои се комбинација од 2 или повеќе алгоритми и/или податочни структури • Наставникот наведува и објаснува решенија на примери од категории проблеми кои се комбинација од 2 или повеќе алгоритми и/или податочни структури • Учениците имплементираат решенија за дадените пример проблеми • Учениците решаваат проблеми кои се сведуваат на изучените категории на проблеми кои се 	<p>Ученикот/ученичката може да:</p> <p>1.1: го објасни пристапот на декомпозиција на проблеми кои се комбинација од 2 или повеќе алгоритми и/или податочни структури;</p> <p>1.2: наброи категории на проблеми кои се комбинација од 2 или повеќе алгоритми и/или податочни структури;</p> <p>1.3: имплементира програмски решенија за изучените пример проблеми кои се комбинација од 2 или повеќе алгоритми и/или податочни структури;</p> <p>1.4: изработува програмски решенија за проблеми кои се комбинација од 2 или повеќе алгоритми и/или податочни структури.</p>

*Внесени се стандарди/индикатори за постигнување на резултатите од учењето врз основа на кои се определуваат критериумите за оценување.

		<p>граф и динамичко програмирање</p> <ul style="list-style-type: none"> - Двокомпонентен проблем кој вклучува граф - Двокомпонентен проблем кој вклучува математика - Двокомпонентен проблем: исцрпно пребарување и геометрија - Двокомпонентен проблем кој вклучува ефикасна податочна структура - Трикомпонентни проблеми • Решавање на избрани проблемски задачи кои ги отсликуваат разгледаните типови <p>Поими: Повеќе компонентни проблеми</p>	<p>комбинација од 2 или повеќе алгоритми и/или податочни структури</p> <p>Методи: дискусија, дијалог, учење преку откривање-истражување, практична работа на компјутер - решавање на програми</p>	
--	--	---	--	--

Модуларна единица 7: СПЕЦИФИЧНИ ПРОБЛЕМСКИ ЗАДАЧИ (24 ЧАСА)

Ред. број	Резултати од учење	Содржини и поими	Активности и методи	Критериуми на оценување*
1	<p>Ученикот/ученичката ќе биде способен/а:</p> <ul style="list-style-type: none"> - Да дискутира и анализира алгоритми/проблемски задачи кои поретко се среќаваат во компетитивното програмирање; - да креира програмски решенија за специфични проблеми од компетитивното програмирање 	<ul style="list-style-type: none"> • Примери на алгоритми/проблемски задачи кои ретко се среќаваат во компетитивното програмирање - Техника на лизгачки прозорец - Проблем на кинески поштар - Проблем на најблизок пар - Најнизок заеднички предок во дрво 	<p>Активности:</p> <ul style="list-style-type: none"> • Наставникот преку презентација и дискусија, со наведување на примери ги запознава учениците со можноста за среќавање со алгоритми/проблемски задачи кои ретко се среќаваат во компетитивното програмирање • Наставникот наведува и објаснува решенија на примери од 	<p>Ученикот/ученичката може да:</p> <p>1.1: објаснува алгоритми/проблемски задачи кои се среќаваат во компетитивното програмирање</p> <p>1.2: имплементира решенија за дадените пример проблеми;</p> <p>1.3: идентификува специфични проблеми од интерес од областа на компетитивни задачи;</p> <p>1.4: изработува програмски решенија за избрани проблемски задачи.</p>

*Внесени се стандарди/индикатори за постигнување на резултатите од учењето врз основа на кои се определуваат критериумите за оценување.

		<ul style="list-style-type: none"> - Варијанти на проблеми со палиндром - „Палачинка“-сортирање • Решавање на избрани проблемски задачи кои ги отсликуваат разгледаните типови • Истражување за проблемски задачи од интерес за учениците и креирање на листа со консензус • Дискутирање на решенија и решавање на избраните проблемски задачи <p>Поими: Техника на лизгачки прозорец, Проблем на кинески поштар, Проблем на најблизок пар, Најнизок заеднички предок во дрво, палиндром, „Палачинка“-сортирање</p>	<p>алгоритми/проблемски задачи кои ретко се среќаваат во компетитивното програмирање</p> <ul style="list-style-type: none"> • Учениците имплементираат решенија за дадените пример проблеми • Учениците решаваат проблеми кои се сведуваат на изучените пример проблеми • Учениците истражуваат за проблемски задачи од нивен интерес и заедно со наставникот креираат листа од задачи за натаможно разгледување • Дискусија на решенија на избраните задачи • Учениците ги имплементираат решенијата на избраните проблемски задачи <p>Методи: дискусија, дијалог, учење преку откривање-истражување, практична работа</p>	
--	--	---	---	--

			на компјутер - решавање на програми	
--	--	--	-------------------------------------	--

Оценување на постигањата на учениците	<p>Во текот на наставата по напредно програмирање се препорачува формативно следење кое вклучува изработка и водење портфолио на учениците што опфаќа:</p> <ul style="list-style-type: none">- собирање 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. <p>На крајот на секое тримесечје, врз основа на сознанијата од формативното оценување, се реализира микросумативното оценување.</p> <p>Согласно природата на програмата по предметот напредно програмирање оценувањето може да се реализира усно, практично, со презентација и слично.</p> <p>Ученикот се оценува со бројчана оценка.</p> <p>Наставникот, според своето согледување, може да го проверува знаењето со усни одговори на учениците, со тестови според модуларните единици, домашни задачи и друго.</p>
--	--

Литература и други извори	<ul style="list-style-type: none"> - Учебник по програмирање избран на ниво на училиштето и одобрен од страна на министерот за образование и наука; - Збирки задачи; - Интернет, образовни софтвери и ресурси; - интегрирана околина за програмирање и демо програми; - аудио-визуелни средства.
Почеток на имплементација на наставната програма	Учебна 2022/2023 година
Институција/ носител на програмата	Биро за развој на образованието(БРО)
Потпис и датум на донесување на наставната програма	<p>бр. 13-7336/10 22.6.2022 година</p> <p style="text-align: right;">МИНИСТЕР, Doc.Dr. Jeton Shaqiri</p> <hr style="width: 20%; margin-left: auto; margin-right: 0;"/>
Датум на ревизија	