Уеб програмиране – Задачи за упражнение

1. Създайте уеб страница за представяне на специалност Информатика.

Добавете следните страници, като за всяка от тях добавите подходящо заглавие.

* Начална страница;
* Обща характеристика на специалността;
* Компетентности на завършилите;
* Перспективи за реализация;
* Гъвкавост на обучението;
* Условия за прием;
* Учебен план;
* Новини;
* Мнения;
* Контакти;
* Полезни връзки.

1.1. Като използвате подходящи HTML елементи добавете следното текстово съдържание към страниците:

* Начална страница

Специалността „Информатика” е от областта на висше образование 4. Природни науки, ма­те­ма­тика и *ин­форматика* по про­­фесионалното направление 4.6. Информатика и компютърни на­у­ки. Обу­­чението по специалността за об­ра­зо­ва­телно-квалификационна сте­пен „Бака­ла­вър” има ре­дов­на и задочна форма с продължителност 4 години. За­­­вършилите ба­ка­лав­ри по тази специалност при­­до­би­­ват про­фесионална квалификация „Информатик“.

* Обща характеристика на специалността

Специалността си поставя за цел да акцентира не само върху езиците и технологиите за програмиране, но и да покаже пътища за анализиране и решаване на разнообразни проблеми във областта софтуерното производство, компютърната графика, изкуствения интелект, мрежи, разпределени приложения, качество и сигурност на софтуера.    Специалността осигурява базова широкопрофилна подготовка по информатика чрез за­дъл­жи­телните фундаментални дисциплини, включени в учебния план. С избираемите дисциплини сту­ден­тите се подготвят по-задълбочено в избрана от тях област в информатиката и най-съвременните софтуерни технологии, а с факултативните дис­циплини могат да изучават специализирани курсове в областта на математиката, икономиката, биз­не­са, правото, психологията, езиковото обучение и други. Практиката по специалността и преддипломният стаж се провеждат в сътрудничество с авторитетните работодателски организации БАСКОМ и „Клъстер информационни и комуникационни технологии Пловдив”. Те осигуряват необходимата подготовка на студентите за реализация на трудовия пазар.

Завършилите специалността притежават базови математически познания, необходими на професионалните информатици в об­ласт­та на линейна алгебра, математически анализ, геометрия, дискретна математика, модели на реални процеси, вероятности и приложна статистика, числени методи. Те имат задълбочени познания върху основите на информатиката и информационните процеси и модели, анализа и прилагането на програмни и информационни технологии, проблемите на информационното обслужване, съвременните методи и системи за проектиране, разработване и внедряване на програмни продукти. Освен необходимите научни познания, учебният план предвижда при­добиване на основните за професионалните информатици технически умения. В процеса на обу­че­ние те са придобили и допълнителни умения и знания като самостоятелно търсене на необходима ин­фор­мация в литературни източници и в Интернет, мултимедийно презентиране на различни проекти, док­лади и съобщения, работа в екип, достатъчно добро владеене на английски език в областта на изучаваната спе­ци­ал­ност.

* Компетентности на завършилите;

При завършване на бака­ла­вър­ска­­та сте­пен на специалността студентът трябва да е способен:

* да показва задълбочени познания за информационните процеси, модели и сис­теми, анализа и прилагането на програмни и ин­фор­ма­ционни технологии;
* да внедрява и използва програмни продукти и информационни системи в различни области на човешката дейност;
* да прилага най-новите информационни технологии в различни сфери от обществения жи­вот;
* да използва математическия апарат за създаване на формални модели в раз­лич­ни пред­мет­ни области;
* да моделира, проектира и създава софтуерни приложения.

Наред с информатичната и математическа компетентности, характерни за професионалното нап­рав­ление, всеки дипломиран информатик тряб­ва да е придобил и следните ключови умения и ком­пе­тен­тности:

* Способност за логическо мислене и моделиране на реални процеси.
* Способност за работа в екип.
* Способност за самостоятелно търсене на информация и натрупване на допълнителни знания.
* Способност за прилагане на математическия апарат в различни предметни области с помощта на най-новите информационни технологии.
* Познания за специфични програмни езици или софтуер.
* Познания за съвременен софтуерен дизайн или уеб-базирани и мобилни системи.
* Умения за създаване на софтуерни приложения.
* Умения за професионална комуникация на английски език.
* Перспективи за реализация

Завършилите специалността са подготвени да прилагат знанията си в различни области на човешката дейност. Бакалаврите по информатика се реализират като спе­ци­а­лис­ти в компютърни и софтуерни фирми, научни организации, банки, застрахователни дружества и дру­ги, където проектират, създават, оценяват и внедряват различните софтуерни системи, обучават пот­ребителите на такива системи и прилагат информационните технологии в различни области на чо­веш­ката практика. След завършване и на магистърска степен те имат възможност да продължат обу­чението си за образователната и научна степен „Доктор” и да се реализират като пре­по­да­ва­те­ли и научни работници във висши училища и научни институти.

* Гъвкавост на обучението;

Във ФМИ са осигурени параметрите, га­ран­­ти­ращи гъвкавостта на избраната специалност: придобиване на допълнителна професионална ква­лификация „Учител по математика”, „Учител по информатика и информационни технологии”; при­добиване на втора или на нова специалност в рамките на обучението по една образователна сте­пен; преминаване от обу­че­ние в една образователна степен към обучение в друга; придобиване на по-висока образователна степен; завършване на обучението в съкратени срокове при желание от страна на обучаваните; придобиване на нова специалност; частично обучение на български и чуж­де­стран­ни студенти в избраната от тях специалност във Факултета; преместване на студенти в друго вис­ше училище, факултет, специалност и форма на обучение; прекъсване на обучението от страна на сту­дентите, както и продължаването му след това; избор на преподавател от студентите, когато по една дисциплина обучението се води от повече от един преподавател.

* Условия за прием;

Приемът за всички бакалавърски специалности във Факултета се реализира на базата на [следната формула](http://fmi.uni-plovdiv.bg/index.jsp?id=866&ln=1).

Знанията, уменията и способностите, които се изискват на конкурсните изпити са дадени в про­грамите за кандидатстудентските изпити по математика и по информатика. Тази информация при­съства в кандидатстудентския справочник на ПУ, както и на [сайта на Факултета](http://fmi.uni-plovdiv.bg/index.jsp?id=317&ln=1).

1.2. Добавете следните хипервръзки:

* В страница Полезни връзки – към сайта на ПУ „Паисий Хилендарски“ и към сайта на ФМИ.
* В страница Условия за прием – към формулата за изчисляване на кандидатстудентски бал и към специалностите, по които се предлага обучение във ФМИ.

1.3. В страница Учебен план добавете следните таблици:

**УЧЕБЕН ПЛАН**

|  |  |
| --- | --- |
| Област на висше образование: | **ПРИРОДНИ НАУКИ, МАТЕМАТИКА И ИНФОРМАТИКА (4)** |
| Професионално направление: | **ИНФОРМАТИКА И КОМПЮТЪРНИ НАУКИ** **(4.6)** |
| Специалност: | **ИНФОРМАТИКА** |
| Образователно- квалификационна степен: | **БАКАЛАВЪР** |
| Професионална квалификация: | **ИНФОРМАТИК** |
| Форма на обучение: | **ЗАДОЧНА** |
| Продължителност на обучението: | **4 ГОДИНИ** |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Дисциплини | А | Л | Лс | У | Ус | Лб | Лбс | СП | О | К | ФИ | Г | Тр | Н |
| ПЪРВА АКАДЕМИЧНА ГОДИНА | 275 | 115 | 11,5 | 90 | 9 | 70 | 7 | 1525 | 1800 | 60 |  |  |  |  |
| Въведение в компютърните науки (C++) | 30 | 15 | 1,5 | 0 | 0 | 15 | 1,5 | 180 | 210 | 7 | И | 1 | Е | 1 |
| Линейна алгебра и аналитична геометрия | 40 | 20 | 2 | 20 | 2 | 0 | 0 | 200 | 240 | 8 | И | 1 | Е | 1 |
| Английски език | 25 | 0 | 0 | 25 | 2,5 | 0 | 0 | 125 | 150 | 5 | - | 1 | Е | 1 |
| Информационни технологии | 30 | 15 | 1,5 | 0 | 0 | 15 | 1,5 | 180 | 210 | 7 | И | 1 | З | 2 |
| Програмиране (C++) | 30 | 15 | 1,5 | 0 | 0 | 15 | 1,5 | 180 | 210 | 7 | И | 1 | З | 2 |
| Английски език | 25 | 0 | 0 | 25 | 2,5 | 0 | 0 | 125 | 150 | 5 | ТО | 1 | З | 2 |
| Операционни системи | 25 | 15 | 1,5 | 0 | 0 | 10 | 1 | 155 | 180 | 6 | И | 1 | П | 3 |
| Програмиране на Java | 30 | 15 | 1,5 | 0 | 0 | 15 | 1,5 | 180 | 210 | 7 | И | 1 | П | 3 |
| Математически анализ | 40 | 20 | 2 | 20 | 2 | 0 | 0 | 200 | 240 | 8 | И | 1 | П | 3 |

1.4. В страницата за контакти добавете форма за изпращане на съобщение с текстови полета за въвеждане на име и фамилия и въпрос, както и бутон за изпращане на съобщението.

1.5. Добавете следното изображение в началото на всички създадени страници:



1.6. Направете меню за връзка между създадените страници и към оформлението на страниците добавите контейнери. Следвайте следната схема:

|  |
| --- |
| #header (ФМИ-изображение) |
| #nav |
| #content (Съдържание на съответната страници) |

1.7. Като използвате SVG в страницата Начало добавете правоъгълник с текст ФМИ в избрани от вас цветове.

1.8. Добавете външен CSS стил към създадените страници.

1.9. Добавете следните стилови формати:

* Header – широчина 90%, височина 10%, отстъп 5%;
* Nav – широчина 90%, височина 10%, отстъп 5%;
* Контейнер content – ширина 90%, височина 80%, отстъп 5%, бял фон;
* Заглавия – централно подравняване, шрифт Arial, размер на шрифта 14 pt, главни букви, удебелен и наклонен шрифт;
* Параграфи – двустранно подравняване, отстъп на първи ред 40px, размер на шрифта 12;
* Фон на страницата – светло сиво;
* Връзки от менюто:
  + Непосетен линк, посетен линк – бял цвят, тъмносин фон;
  + Активен линк – тъмносин цвят, бял фон.
* Връзки в съдържанието на страниците:
  + Непосетен линк – тъмно син цвят;
  + Активен линк – бял цвят, тъмносин фон;
  + Посетен линк – тъмносин цвят, неподчертан;
  + Посочен линк – тъмносин цвят, удебелен шрифт.
* Форматиране на списъци – водещ символ квадратче, отстъп 40px, шрифт Arial;
* Таблица с учебен план:
  + Заглавен ред – централно подравняване, удебелен шрифт, тъмносин фон, бял цвят;
  + Ред за академична година – централно подравняване, удебелен шрифт, светлосив фон, черен цвят;

2. Да се създаде програма, която изисква от потребителя да въведе две числа в диалогови прозорци prompt и намира средното аритметично на въведените числа.

<HTML>

<HEAD>

<SCRIPT LANGUAGE=JavaScript><!--

var num1;

num1=prompt("Въведи първо число.","");

num1=parseFloat(num1);

var num2;

num2=prompt("Въведи второ число.","");

num2=parseFloat(num2);

var avg;

avg=(num1+num2)/2;

alert("Средното аритметично на " + num1 + " и " + num2 + " е " + avg);

//--></SCRIPT>

</HEAD>

<BODY>

<P>

</BODY>

</HTML>

3. Създайте JavaScript програма, която:

* При отваряне на страницата се показва диалогов прозорец confirm, който задава на потребителя въпроса "Are you sure?". Диалоговият прозорец съдържа два бутона: "OK" и "Cancel".
* Ако потребителят натисне "Cancel" програмата не прави нищо.
* Ако потребителят натисне "OK" програмата прави следното:
  + - Показва диалогов прозорец prompt, в който потребителят трябва да въведе пълното си име.
    - Показва диалогов прозорец prompt, в който потребителят трябва да въведе ден от седмицата.
    - Показва диалогов прозорец alert, показващ съобщение подобно на следното "Dear XXX, we are so happy to see you on such a wonderful YYY as today!", където XXX и YYY са въведените от потребителя име и ден от седмицата.

Запишете вашата работа в HTML файл.

<html>

<body>

<script>

var r=confirm("Are you sure");

if (r==true)

{

var name=prompt("What is your first name?", "");

var day=prompt("Day?", "")

if (name!=null && day!=null)

alert("Dear "+name+", we are so happy to see you on such a wonderful "+day+" as today!");

}

else

{

x="You pressed Cancel!";

}

</script>

</body>

</html>

4. Да се създаде програма, която намира сумата на числата от 1 до въведено от потребителя число. Да се използва цикъл for и да се проверява дали въведеното число не е по-малко от 1.

<html>

<body>

<script language=JavaScript>

var n;

do

{

n=parseInt(prompt("Въведи число",1));

}

while (n<1);

var sum=0;

for (var i=1;i<=n; i++)

{

sum=sum+i;

}

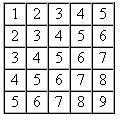
alert("Сумата на числоата от 1 до " + n + " e " +sum);

</script>

</body>

</html>

5. Да се създаде Javascript програма, която отпечатва следната таблица:



<script>

document.write("<table border='2'>")

for (var i=1; i<=5; i++)

{document.write("<tr>")

for (j = i; j <= i+4; j++)

{document.write("<td>" + j + "</td>")}

document.write("</tr>")

}

document.write("</table>")

</script>

6. Двоично търсене: Напишете уеб-базирана програма, която реализира играта “познай числото". Числото е произволно цяло число между 1 и 999. След това дайте на потребителя до 10 опита. След всеки опит визуализирайте съобщение "Too Low", "Too High" или "Got it!".

<!DOCTYPE html>

<html>

<head>

<script>

function guess(x,y,number)

{

do

{

var s=prompt("Guess the number between "+x+" and "+y,"");

}

while (s<x || s>y)

var s1=parseInt(s);

if (s1==number)

{alert("You win the game");}

else

{if (number>s1)

{

alert( s1 + " is too low number. Try again!");

guess(s1,y,number);

}

else

{

alert( s1 + " is too big number. Try again!");

guess(x,s1,number);

}

}

}

</script>

</head>

<body>

<h1>Guess the number</h1>

<script>guess(1,999,Math.floor(Math.random()\*999))</script>

</body>

</html>

<!DOCTYPE html>

<html>

<head>

<meta charset=utf-8 />

<title>JS Bin</title>

</head>

<body>

<script language=JavaScript>

var QUIZZES=5;

do

{

var num=prompt("въведи брой студенти");

}

while (isNaN(num) || num<1);

var grades=new Array(num);

for (var student=0; student<num; student++)

{

grades [student] = new Array(QUIZZES);

for (var gr=0; gr<QUIZZES; gr++)

{

do

{

var grade=prompt("Въведи оценка на студент " +

(student+1) + " по предмет" +(gr+1),2);

}

while (isNaN(grade) || grade<2 || grade>6);

grades[student][gr]=parseInt(grade);

}

}

for (student=0; student<grades.length; student++)

{ sum=0;

for (gr=0; gr<QUIZZES; gr++)

{sum+=grades[student][gr];}

average=sum/QUIZZES;

document.write("Оценките на студент "+(student+1)+"

са "+grades[student].toString()+"</br>");

document.write("Средният успех на студент " + (student + 1) +

" е " + average+"</br>");

}

</script>

</body>

</html>

7. Да се напише програма, в която от потребителя се изисква да въведе чрез диалогови прозорци името и фамилията си. При въвеждане на данните да се проверя дали въведеното име съдържа поне 3 символа, а фамилията 5 символа. След това да се извършат следните действия и резултатите да се отпечата на екрана:

* + Въведените данни да се съхранят в низ, като за разделител се използва интервал;
  + Да се намери дължината на низа;
  + Първите букви на името и фамилията да се визуализират с главни букви:
  + Всички символи на низа да се визуализират с главни букви.

<!DOCTYPE html>

<html>

<head>

<meta charset=utf-8 />

<title>JS Bin</title>

</head>

<body>

<script language=JavaScript>

do

{

var name=prompt("Въведи име", "");

}

while (name.length<3);

do

{

var family=prompt("Въведи фамилия", "");

}

while (family.length<5);

var str=name.concat(" ",family);

document.writeln(str +"</br>");

document.writeln(str.length +"</br>");

var strsmall=str.toLowerCase();

document.writeln(strsmall +"</br>");

var strreplacefirst= str.replace(str.charAt(0), str.charAt(0).toUpperCase());

var n=str.indexOf(" ");

var strreplace= strreplacefirst.replace(str.charAt(n+1), str.charAt(n+1).toUpperCase());

document.writeln(strreplace +"</br>");

document.writeln(str.toUpperCase());

</script>

</body>

</html>

8. Създайте уебстраница, която ви предлага как да играете TOTO.

<!DOCTYPE html>

<html>

<head>

<script language="javascript">

function Toto()

{

var num=new Array()

for (var i=0; i<6;i++)

{var add = true;

number=Math.floor(Math.random()\*49+1);

for(j = 0; j < 6; j++)

{

if(num[j] == number){add = false;}

}

if(add){num.push(number);}

else

{ i--;}

}

document.write(num);

}

</script>

</head>

<body>

How to play Toto 6/49<br>

<script>Toto()</script>

</body>

</html>

9. Напишете програма, която при натискане на бутон да пресмята стойността на математическа функция според стойност въведена в текстово поле и да връща резултата в друго текстово поле. Използвайте радио бутони, чрез които потребителят да избира какво да се пресмята, напр. корен квадратен, синус, косинус, и т.н.

<html>

<head>

<script type="text/javascript">

function compute()

{

var s=document.getElementById("form1").value;

var r=parseFloat(s);

if(document.forms[0].Choice[0].checked)

r=r\*r;

else

r=Math.sqrt(r);

s=String(r);

document.getElementById("form2").value=s;

}

</script>

</head>

<body>

Input an integer:

<input type="text" id="form1" value="0" size="20">

<br>What you want to compute:

<form>

<input type="radio" checked="checked" name="Choice" value="Square">

Square <br>

<input type="radio" name="Choice" value="SquareRoot">

Square Root:

</form>

<input type="submit" id="button1" value="Click to compute" onclick="compute()">

<br>The result is:

<input type="text" readonly id="form2" value="0" size="20">

</body>

</html>

10. Напишете код, който проверя валидността на ЕГН. Поискайте от потребителя да въведе нейния/неговия пол и датата на раждане. Използвайте следния критерий: пресмята се остатъка от делението на сумата на произведенията от позицията на i-тата цифра ( i=1,...,9,) със съответното тегло (2,4,8,5,10,9,7,3,6) с 11 и се проверява дали получената стойност е равна на последната цифра от ЕГН (за подробности вж. <http://www.grao.bg/esgraon.html>)

<!DOCTYPE html>

<html>

<head>

<script type="text/javascript">

function checkegn()

{

var s=document.forms[0].egn.value;

if (s.length!=10) alert("invlid input")

var numbers=s.split("",10);

var arr=new Array(2,4,8,5,10,9,7,3,6);

var i;

var count=0;

for (i=0; i<10; i++)

{ if (isNaN(numbers[i])) count++;}

if (count>0) alert("EGN must contains only numbers");

var sum=0;

for (i=0;i<9;i++)

{sum=sum+numbers[i]\*arr[i];}

var compute=sum%11;

if (compute<10)

controlnumber=compute;

else

controlnumber=0;

if (controlnumber==numbers[9]) alert("Valid EGN")

else alert("invalid EGN")

}

</script>

</head>

<body>

<form>

Insert EGN:

<input type="text" id="egn" size="20" value="0"><br>

<input type="button" id="button1" value="Check" onclick="checkegn()">

</form>

</body>

</html>