

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ

імені ІГОРЯ СІКОРСЬКОГО”

Факультет прикладної математики

Кафедра програмного забезпечення комп’ютерних систем

**Розрахункова робота**

з дисципліни “Програмне забезпечення систем автоматичної ідентифікації”

Варіант №5

|  |  |  |  |
| --- | --- | --- | --- |
| Виконав  студент V курсу  групи КП-11мн  Кравчук Аркадій Андрійович |  |  | Перевірила  “\_\_\_\_” “\_\_\_\_\_\_\_\_\_\_\_\_” 2021 р.  викладач  Боярінова Юлія Євгеніївна |

Київ 2021

**Мета роботи**

Набуття студентами практичного досвіду з розроблення штрихових кодів з оптимізованими параметрами.

**Завдання**

Варіант №5 за списком.

Розробити спосіб штрихового кодування блоку алфавітно-цифрових даних про одиницю обліку з підвищеною інформаційною щільністю за допомогою штрихового коду EAN.

**Код програмного забезпечення**

Лістинг програмного модулю для кодування штрих-кодової позначки:

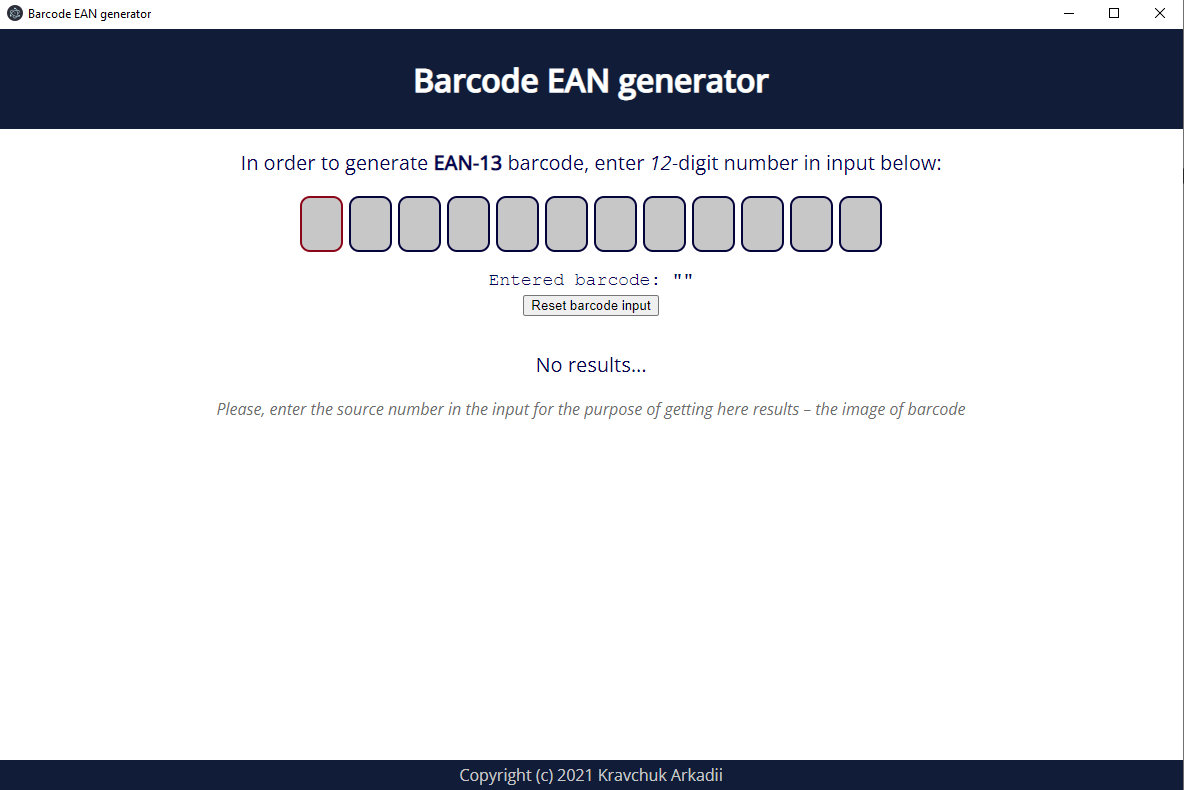
|  |
| --- |
| // EAN13  class EAN13 {  private static H1 = [1, 0, 1];  private static H4 = [0, 1, 0, 1, 0];  private static A = [  [0, 0, 0, 1, 1, 0, 1],  [0, 0, 1, 1, 0, 0, 1],  [0, 0, 1, 0, 0, 1, 1],  [0, 1, 1, 1, 1, 0, 1],  [0, 1, 0, 0, 0, 1, 1],  [0, 1, 1, 0, 0, 0, 1],  [0, 1, 0, 1, 1, 1, 1],  [0, 1, 1, 1, 0, 1, 1],  [0, 1, 1, 0, 1, 1, 1],  [0, 0, 0, 1, 0, 1, 1]  ];  private static B = [  [0, 1, 0, 0, 1, 1, 1],  [0, 1, 1, 0, 0, 1, 1],  [0, 0, 1, 1, 0, 1, 1],  [0, 1, 0, 0, 0, 0, 1],  [0, 0, 1, 1, 1, 0, 1],  [0, 1, 1, 1, 0, 0, 1],  [0, 0, 0, 0, 1, 0, 1],  [0, 0, 1, 0, 0, 0, 1],  [0, 0, 0, 1, 0, 0, 1],  [0, 0, 1, 0, 1, 1, 1]  ];  private static C = [  [1, 1, 1, 0, 0, 1, 0],  [1, 1, 0, 0, 1, 1, 0],  [1, 1, 0, 1, 1, 0, 0],  [1, 0, 0, 0, 0, 1, 0],  [1, 0, 1, 1, 1, 0, 0],  [1, 0, 0, 1, 1, 1, 0],  [1, 0, 1, 0, 0, 0, 0],  [1, 0, 0, 0, 1, 0, 0],  [1, 0, 0, 1, 0, 0, 0],  [1, 1, 1, 0, 1, 0, 0]  ];  private static first\_half = [  [EAN13.A, EAN13.A, EAN13.A, EAN13.A, EAN13.A, EAN13.A],  [EAN13.A, EAN13.A, EAN13.B, EAN13.A, EAN13.B, EAN13.B],  [EAN13.A, EAN13.A, EAN13.B, EAN13.B, EAN13.A, EAN13.B],  [EAN13.A, EAN13.A, EAN13.B, EAN13.B, EAN13.B, EAN13.A],  [EAN13.A, EAN13.B, EAN13.A, EAN13.A, EAN13.B, EAN13.B],  [EAN13.A, EAN13.B, EAN13.B, EAN13.A, EAN13.A, EAN13.B],  [EAN13.A, EAN13.B, EAN13.B, EAN13.B, EAN13.A, EAN13.A],  [EAN13.A, EAN13.B, EAN13.A, EAN13.B, EAN13.A, EAN13.B],  [EAN13.A, EAN13.B, EAN13.A, EAN13.B, EAN13.B, EAN13.A],  [EAN13.A, EAN13.B, EAN13.B, EAN13.A, EAN13.B, EAN13.A],  ];  private source\_number: string;  private source\_number\_arr: number[];  private check\_digit: number;  private barcode\_arr: number[];  public constructor(source\_number: string) {  this.source\_number = source\_number;  this.source\_number\_arr = [];  for (let i = 0; i < source\_number.length; i++) {  this.source\_number\_arr.push(parseInt(source\_number[i]));  }  this.check\_digit = this.calculateCheckDigit();  this.barcode\_arr = this.calculatBarcode();  }  private calculatBarcode() {  const res: number[] = [];  const check\_digit = this.check\_digit;  const cur\_first\_half = EAN13.first\_half[check\_digit];  for (let i = 0; i < EAN13.H1.length; i++) {  res.push(EAN13.H1[i]);  }  for (let i = 12; i > 6; i--) {  const cur\_digit\_arr = cur\_first\_half[12 - i];    const cur\_digit = this.source\_number\_arr[12 - i];  const vals\_for\_digit = cur\_digit\_arr[cur\_digit];  for (let j = 0; j < vals\_for\_digit.length; j++) {  res.push(vals\_for\_digit[j]);  }  }  for (let i = 0; i < EAN13.H4.length; i++) {  res.push(EAN13.H4[i]);  }  for (let i = 6; i < 12; i++) {  const cur\_digit = this.source\_number\_arr[i];  const vals\_for\_digit = EAN13.C[cur\_digit];  for (let j = 0; j < vals\_for\_digit.length; j++) {  res.push(vals\_for\_digit[j]);  }  }  for (let i = 0; i < EAN13.H1.length; i++) {  res.push(EAN13.H1[i]);  }  return res;  }  private calculateCheckDigit() {  let odd\_sum = 0;  let even\_sum = 0;  for (let i = 0; i < this.source\_number\_arr.length; i++) {  if (i % 2 == 0) {  even\_sum += this.source\_number\_arr[i];  } else {  odd\_sum += this.source\_number\_arr[i];  }  }  const sum = (even\_sum \* 33) + odd\_sum;  return (10 - (sum % 10)) % 10;  }  public getSourceNumber() {  return this.source\_number;  }  public getBarcodeArr() {  return this.barcode\_arr;  }  public getCountryCode() {  return this.source\_number.substr(0, 3);  }  public getManufacturerCode() {  return this.source\_number.substr(3, 4);  }  public getProdcutCode() {  return this.source\_number.substr(7, 5);  }  public getCheckDigit() {  return this.check\_digit;  }  }  export default EAN13; |

Лістинг програмного модулю для відображення штрих-кодової позначки:

|  |
| --- |
| import React from "react";  import DigitsPanel from "./digits";  import EAN13 from "./ean13";  const BARCODE\_DIGITS\_COUNT = 12;  const BARCODE\_NAME = "EAN-13";  type AppState = { barcode\_source\_number\_val: string, barcode\_source\_number\_is\_valid: boolean, barcode\_obj: EAN13 | null };  class App extends React.Component {  state: AppState = {  barcode\_source\_number\_val: "",  barcode\_source\_number\_is\_valid: false,  barcode\_obj: null,  }  remove\_focus\_from\_last\_digit\_if\_needed() {  const last\_digit = document.getElementById("last-input-digit");  if (document.activeElement === last\_digit) {  last\_digit.blur();  }  }  check\_if\_barcode\_source\_number\_val\_is\_valid(val: string) {  return val.length == BARCODE\_DIGITS\_COUNT && /^[0-9]{12}$/.test(val);  }  handleChange\_barcode\_source\_number\_val = (val: string) => {  const is\_valid = this.check\_if\_barcode\_source\_number\_val\_is\_valid(val);  this.setState({  barcode\_source\_number\_val: val,  barcode\_source\_number\_is\_valid: is\_valid  });  if (is\_valid) {  this.setState({  barcode\_obj: new EAN13(val)  });  this.remove\_focus\_from\_last\_digit\_if\_needed();  }  }  reset\_barcode\_source\_number\_val = () => {  this.setState({  barcode\_source\_number\_val: "",  barcode\_source\_number\_is\_valid: false  });  this.make\_focus\_on\_first\_digit\_if\_needed();  }  make\_focus\_on\_first\_digit\_if\_needed() {  const first\_digit = document.getElementById("first-input-digit");  if (document.activeElement !== first\_digit) {  first\_digit.focus();  }  }  append\_results\_block\_if\_needed = () => {  if (this.state.barcode\_source\_number\_is\_valid) {  return (  <div>  <hr style={{ margin: "0 30px" }}/>  <p style={{ textAlign: "center", marginTop: "10px", fontSize: "20px", fontWeight: 500 }}>  Results:  </p>  <canvas width="190px" height="95px" id="barcode-canva" style={{ display: "block", marginLeft: "auto", marginRight: "auto", marginBottom: "20px" }}></canvas>  <textarea readOnly={true} value={this.state.barcode\_obj.getBarcodeArr().join("")} style={{ fontFamily: "monospace", width: 760, display: "block", marginLeft: "auto", marginRight: "auto" }}>  </textarea>  <pre style={{ textAlign: "center" }}>Check digit: {this.state.barcode\_obj.getCheckDigit()}</pre>  </div>  );  } else {  return (  <div style={{ display: "flex", flexDirection: "column", alignItems: "center", justifyContent: "center", alignContent: "center" }}>  <p style={{ textAlign: "center", fontSize: "20px" }}>No results...</p>  <p style={{ textAlign: "center", marginTop: "0px", fontStyle: "italic", color: "GrayText" }}>  Please, enter the source number in the input for the purpose of getting here results – the image of barcode  </p>  </div>  );  }  }  componentDidUpdate() {  if (this.state.barcode\_source\_number\_is\_valid) {  const canvas = document.getElementById('barcode-canva') as HTMLCanvasElement;  if (canvas && canvas.getContext) {  const barcode\_arr = this.state.barcode\_obj.getBarcodeArr();  const ctx = canvas.getContext('2d');  const i\_width = 2;  const height = 70;    ctx.clearRect(0, 0, i\_width \* 95, height + 15);  for (let i = 0; i < barcode\_arr.length; i++) {  if (barcode\_arr[i] == 1) {  let tmp\_height = height;  if ((i >= 0 && i <= 2) || (i >= 45 && i <= 49) || (i >= 92 && i <= 94)) {  tmp\_height += 15;  }  ctx.fillRect(i\*i\_width, 0, i\_width, tmp\_height);  }  }  }  }  }  render() {  return (  <div style={{ height: "calc(100% - 20px)", marginTop: "20px" }}>  <p style={{ marginTop: "0", textAlign: "center", fontSize: "20px"}}>  In order to generate <b>{BARCODE\_NAME}</b> barcode, enter <i>{BARCODE\_DIGITS\_COUNT}</i>-digit number in input below:  </p>  <DigitsPanel count\_digits={BARCODE\_DIGITS\_COUNT} value={this.state.barcode\_source\_number\_val} on\_value\_change={this.handleChange\_barcode\_source\_number\_val} />  <pre className="source-barcode">  Entered barcode: "<span><b>{this.state.barcode\_source\_number\_val}</b></span>"  </pre>  <p className="source-barcode-actions">  <button onClick={this.reset\_barcode\_source\_number\_val}>Reset barcode input</button>  </p>  {this.append\_results\_block\_if\_needed()}  </div>  );  }  }  export default App; |

**Демонстрація програмного забезпечення**

Графічний інтерфейс для кодування штрих-кодової позначки:



Процес кодування штрих-кодової позначки:

