

```

<html><title>Выбрать </title>
<head>

<style>

form {
  padding-top: 50px;
  display: flex;
  justify-content: center;
}
select {
  margin: 5px;
}
</style>

</head>
<body>
<form action="#" method="post">
<select id="select-1">
  <option value="select">Выберите1.</option>
  <option value="a">A</option>
  <option value="b">B</option>
  <option value="c">C</option>
</select>

<select id="select-2">
  <option class="select" value="select">Выберите2</option>
  <option class="a" value="aa">A-A</option>
  <option class="a" value="ab">A-B</option>
  <option class="a" value="ac">A-C</option>
  <option class="b" value="ba">B-A</option>

  <option class="b" value="bb">B-B</option>
  <option class="b" value="bc">B-C</option>
  <option class="c" value="ca">C-A</option>
  <option class="c" value="cb">C-B</option>
  <option class="c" value="cc">C-C</option>
</select>

<select id="select-3">
  <option class="select" value="select">Выберите3</option>
  <option class="aa" value="aaa">A-A-A</option> //здесь ссылки
  <option class="aa" value="aab">A-A-B</option>
  <option class="ab" value="aba">A-B-A</option>
  <option class="ab" value="abb">A-B-B</option>

  <option class="ac" value="aca">A-C-A</option>
  <option class="ac" value="acb">A-C-B</option>
  <option class="ba" value="baa">B-A-A</option>
  <option class="ba" value="bab">B-A-B</option>
  <option class="bb" value="bba">B-B-A</option>
  <option class="bb" value="bbb">B-B-B</option>

  <option class="bc" value="bca">B-C-A</option>
  <option class="bc" value="bcb">B-C-B</option>
  <option class="ca" value="caa">C-A-A</option>
  <option class="ca" value="cab">C-A-B</option>
  <option class="cb" value="cba">C-B-A</option>
  <option class="cb" value="cbb">C-B-B</option>

  <option class="cc" value="cca">C-C-A</option>
  <option class="cc" value="ccb">C-C-B</option>
</select>
</form>
<script>
(() => {
  const dynamicSelect = (id1, id2) => {
    // Определение переменных, ссылающихся на списки
    const sel1 = document.getElementById(id1);
    const sel2 = document.getElementById(id2);
    // Клонирование динамического списка
    const clone = sel2.cloneNode(true);
    // Определение переменных для клонированных элементов списка
    const clonedOptions = clone.getElementsByTagName("option");
  }
});

```

```

// Вызов функции собирающей вызываемый список
refreshDynamicSelectOptions(sel1, sel2, clonedOptions);
// При изменении выбранного элемента в первом списке: // вызов функции пересобирающей вызываемый список
sel1.addEventListener('change', () => {
  refreshDynamicSelectOptions(sel1, sel2, clonedOptions);
});
};

// Функция для сборки динамического списка
const refreshDynamicSelectOptions = (sel1, sel2, clonedOptions) => {
  // Удаление всех элементов динамического списка
  while (sel2.options.length) {
    sel2.remove(0);
  }
  const selectedOption = sel1.options[sel1.selectedIndex].value;
  // Перебор клонированных элементов списка
  for (let i = 0; i < clonedOptions.length; i++) {
    const option = clonedOptions[i];
    // Если название класса клонированного option эквивалентно "select"
    // либо эквивалентно значению option первого списка
    if (option.classList.contains('select') ||
        option.classList.contains(selectedOption)) {
      // его нужно клонировать в динамически создаваемый список
      sel2.appendChild(option.cloneNode(true));
    }
  }
  // Отправляем событие change выбранного select
  const event = document.createEvent('HTMLEvents');
  event.initEvent('change', true, false);
  sel2.dispatchEvent(event);
};

// Вызов скрипта при загрузке страницы
document.addEventListener("DOMContentLoaded", () => {
  dynamicSelect("select-1", "select-2");
  dynamicSelect("select-2", "select-3");
});
})();

</script>
</body>
</html>

```